TDR-64-448
OL. II

FACTORS THAT AFFECT THE COST OF COMPUTER PROGRAMMING

VOLUME II

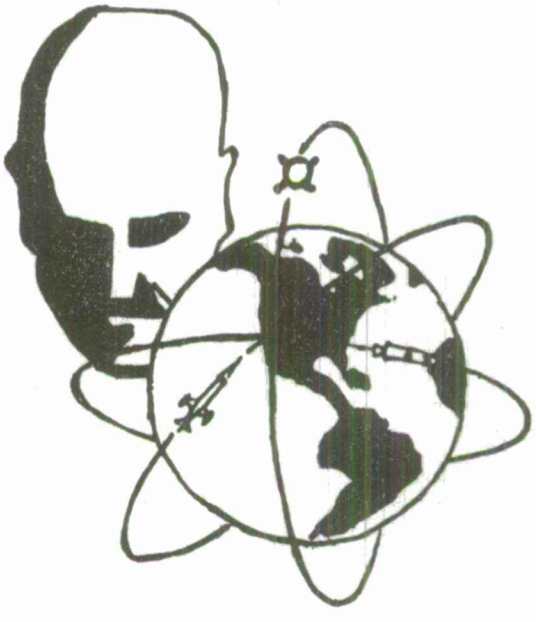
A Quantitative Analysis

TECHNICAL DOCUMENTARY REPORT NO. ESD-TDR-64-448

SEPTEMBER 1964

L. Farr
H. J. Zagorski

DIRECTORATE OF COMPUTERS
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts



(Prepared under Contract No. AF 19 (628)-3418 by the System Development Corporation, Santa Monica, California, 90406.)

DDC AVAILABILITY NOTICES

Qualified requesters may obtain copies from Defense Documentation Center (DDC). Orders will be expedited if placed through the librarian or other person designated to request documents from DDC.

Copies available at Office of Technical Services, Department of Commerce.

ESD-TDR-64-448
VOL. II

FACTORS THAT AFFECT THE COST OF COMPUTER PROGRAMMING

VOLUME II

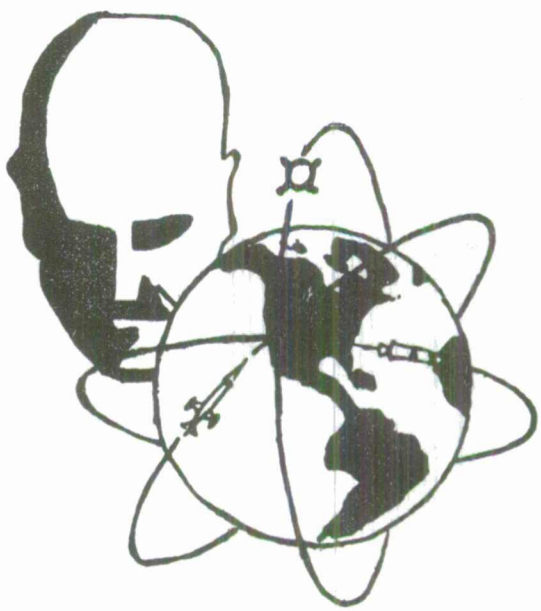
A Quantitative Analysis

TECHNICAL DOCUMENTARY REPORT NO. ESD-TDR-64-448

SEPTEMBER 1964

L. Farr
H. J. Zagorski

DIRECTORATE OF COMPUTERS
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts



(Prepared under Contract No. AF 19 (628)-3418 by the System Development Corporation, Santa Monica, California, 90406.)

FOREWORD

This document is the second of two Technical Documentary Reports prepared for the Air Force Electronic Systems Division as part of a project to develop better techniques for estimating the costs of computer programming.

The first volume of this report describes the work done to identify and to organize the many factors that affect the cost of computer programs. Much of this work that served as a basis for the quantitative analysis was performed during the summer and fall of 1963 under the sponsorship of DOD Advanced Research Projects Agency. The quantitative analysis described in this second volume began on 1 March 1964 under the sponsorship of ESD.

The two volumes of this TDR bear the following System Development Corporation document numbers:

Volume I - TM-1447/000/02

Volume II - TM-1447/001/00

ABSTRACT

Results of an exploratory analysis aimed at deriving better cost-estimating relationships for computer programming development are presented. Based upon previous work that hypothesized an initial list of factors affecting cost, the report describes the steps taken to collect and analyze data for the purpose of supporting or rejecting the presumed factors. As a result, equations that estimate costs in terms of such resources as man months and computer hours have been derived. Since these estimating devices were evolved from a small and, perhaps, unrepresentative sample of programs, the use of these equations is not recommended for actual planning. The study concludes that multivariate regression analysis, supplemented by pertinent judgment and intuition, is an appropriate tool for deriving cost-estimating relationships. To arrive at more useful prediction equations, recommendations are made for continuing the research. These include increasing the sample size and improving the questionnaire used to collect data. The basic inputs for the analyses, the actual cost data, representing twenty-seven program development efforts, are included.

REVIEW AND APPROVAL

This Technical Documentary Report has been reviewed and is approved.

Robert Savoy

ROBERT SAVOY
Project Official

ACKNOWLEDGMENT

The research reported herein was conducted at the System Development Corporation in the Computer Program Implementation Process (CPIP) project. This project is led by Victor LaBolle, whose direction and suggestions for improvements in the work are gratefully acknowledged.

N. E. Willmorth reviewed the document for technical content and provided many stimulating suggestions and criticisms.

Lt. K. Kingston of the U. S. Naval Postgraduate School provided valuable computational assistance while attached to this project as a summer student.

Ann Walker provided fine editorial support; and thanks are extended to Carol Castillo for her painstaking efforts in the typing of many drafts.

TABLE OF CONTENTS

List of Tables

List of Illustrations

I. INTRODUCTION

This is the second of two volumes prepared for the Air Force Electronic Systems Division (ESD) as part of a project to develop better techniques for estimating the costs of computer programming. The general project objective is to conduct research and analysis aimed at developing tools and guidelines for both managers and buyers of computer programming products. These aids are intended to help managers improve the control and planning of computer program development by providing means for lowering costs, shortening lead times, and improving product quality. Additionally, the long-range results of this work are intended to help buyers compare and evaluate computer programming products on a systematic basis.

Since little is known today about cost-estimating relationships for computer program development, both the research and the techniques used to conduct it have been exploratory. The work, therefore, must be iterative in nature. Since the results of this initial analysis have not yielded readily useful tools for managers, the major emphasis in this report is on the approach and methods. This document, therefore, reports on the results of the following activities.

- Definition of cost factors.

  Previous work was reported in the first volume of this series.

- Collection of cost data.

  A questionnaire was designed and used to measure the existence of presumed cost factors in a number of program development efforts.

- Formulation of a prediction model.

  A linear combination of cost factors with appropriately assigned weights was hypothesized as a suitable cost-estimating model.

- Exploration of various statistical techniques that could be used to develop cost-estimating relationships.

  The techniques explored included correlation analysis, regression analysis, and factor analysis, all supplemented by pertinent judgment and intuition.

- Evaluation and documentation of the analysis and results.

  Evaluation in any rigorous sense (e.g., cross-validation with another data sample or actual use) was not possible. However, while the resulting equations are not recommended for actual use in development efforts, the authors would be most anxious for readers to use these equations on an experimental basis. Their reports of success or failure, and reasons for deficiencies in the equations would be extremely valuable.

Section II, Statement of the Problem identifies and discusses the management problem of computer program costing in the context of cost estimation for automatic data processing systems. The requirement for and benefits of accurate cost estimation are cited. This section also serves to define the problem addressed by the study as that of deriving an initial cost estimate for computer program development and does not address the problem of costing program changes.

Section III, Approach and Methods, describes the exploratory research that constitutes the core of this study. The technique of data collection by questionnaire, as used in this analysis, is discussed, with emphasis on some of the problems faced by the investigators. These problems center on the general unreliability and unavailability of computer programming cost data.

The primary analytical technique used in this study was the sequential application of linear multivariate regression analysis, supplemented heavily by pertinent judgment and intuitive analysis. Other techniques used included correlation analysis and factor analysis. Statistical techniques using only available data (survey research) often suffer from two serious problems; both are encountered in this study. One is the lack of control in data collection which results in less than optimum distribution of data (e.g., gaps, skewness); and the second is simply an insufficient number of representative observations. Experience with the techniques described in this section indicated that they are sufficiently robust to supply useful cost-estimating relationships. If more data are collected, the validity and confidence one may place in the resulting equations will be increased.

The resulting estimating equations are described in Section IV, Summary of Results. Illustrative formulas are shown for such costs as man months and computer hours, and product characteristics such as number of delivered program instructions. It is emphasized that the formulas are not sufficiently valid, have large standard errors of estimate, and are primarily illustrative of what could be done with greater quantities of data. The effects of removing three extreme data points are treated in a separate analysis which suggests that different populations may be necessary to describe computer programming development.

The final section, Recommendations for Future Work, outlines several additional techniques that may prove valuable in further analysis and recommends the extensive collection of data, particularly outside of SDC. This is necessary to increase the sample size and to eliminate the potential bias introduced by examining the data of only one organization. Other research highly pertinent to the problem of cost estimation is also recommended. This includes work on techniques for estimating program size and the formulation of descriptors and measures of program performance and quality.

To keep the main body of the report brief, numerical and computational details have been placed in the appendices. These include a copy of the data collection questionnaire, identification of all the variables examined in the study, the responses to the questionnaires, the correlation of each variable with cost (validity table), the results of a preliminary factor analysis, and a summary of the regression analysis for each derived equation. The appendices contain sufficient information to allow independent investigators to repeat any part of the study or to continue it in other desirable directions. In fact, the compilation of cost data included in this report is felt to be the first and most comprehensive collection of its kind, and therefore, a valuable resource.

II. STATEMENT OF THE PROBLEM

Background

The number and size of information-handling systems that use automatic data processing (ADP) have continued to grow in order to support management and operations in both government and industry. Despite the rapid growth in applications of ADP, the development of a reliable technology with a set of principles and techniques for managing programming efforts is noticeably absent. Little effort has been devoted to the collection of data on past experience and to the organization of these data into a systematic body of knowledge for managers. Ad hoc groups organized to examine ADP applications in military command and control operations (e.g., the Air Force Winter Study Group in 1959 and the Institute of Naval Studies Summer Study in 1961) have found that computer program development, in comparison to equipment development, is a lagging technology.

One particularly acute problem in computer program development is that of cost estimation. Recent Congressional hearings concerning the federal use of electronic data processing equipment stress the need for "more specific and systematic measures of cost." General Terhune of the Air Force Electronic Systems Division, in an address to the American Federation of Information Processing Societies (Las Vegas, 1963) stated that "there is no reliable

way to estimate time and costs of initial program jobs." In many cases, cost estimation has been overly optimistic; in others, it has been neglected in planning; as a result, buyers have often been surprised at the real cost of program development. In addition to the problem of initial costing, there is a need to develop techniques for costing changes in the programming project. Although there have been efforts to improve the prediction of equipment costs and lead times, similar work has not kept pace in the programming community.

Another important and related problem is the lack of measures of program performance and quality. When one purchases a hardware component, some statements (usually quantitative) concerning its performance and quality can be made. As a result, both producer and buyer have a means toward a common understanding of the relationship between price, performance and quality. No such means toward a similar understanding exists for the relationship of price, performance and quality in computer programming. The disillusionment of buyers and users, the problems faced by programming managers, and the need to establish more accurate and meaningful cost/value relationships for programs and their development have led to the present need for research into computer programming management.

In answer to this need, a formal research project to investigate problems in programming management was initiated at the System Development Corporation in 1962 by the Advanced Research Projects Agency (ARPA). This project, the Computer Program Implementation Process (CPIP) project, at first sought to determine whether enough similarity existed among various program development efforts to permit analysis of the process of programming in a systematic way. In the early project work, significant similarity in programming was found in terms of the activities that constitute the programming process and the problems that are commonly encountered. To reach this conclusion, project members collected some data on implementation experience, both qualitative and quantitative, in a survey of development efforts by SDC and other organizations. The quantitative data consisted of measures of product size, such as number of pages of documentation, number of program instructions, and costs measured in man months and computer hours.

After identifying a broad range of problem areas, CPIP project members began a more detailed investigation of the factors that contribute to programming costs. In March 1964, ESD contracted with SDC for an extension of the cost analysis to include detailed cost data and appropriate statistical analysis. This report is the result of that work.

The Cost-Estimating Problem

This study was undertaken to explore various techniques to derive estimating relationships for the costs of computer programs. By costs, we mean the resources that are required to produce a program, primarily man months of programmer time and computer hours. To insure that results would be useful

to a large number of managers in different organizations, we did not use dollars as a cost measure because they tend to be influenced by differential wage rates, computer costs and overhead charges.

Good cost estimation is necessary in successful computer program management for many reasons, including the following:

a. Cost estimates serve as the basis for budget-planning decisions. The computer program may be a significant element in the total cost of a command and control system. In the framework of cost effectiveness, a decision to select one system over another may be strongly influenced by the cost of programming. Better cost-estimating techniques would reduce the uncertainty in making such decisions.

b. Cost estimates are used for resource allocation and control. Cost estimates serve as a guideline (in some cases, an upper limit) for the resources that are allocated to the work. Within these limits, resources are apportioned according to the estimates for various parts of the project. While the programming project is in process, the estimates aid in controlling resource expenditure and reallocation. Thus, accurate estimates will improve both allocation and control.

c. Cost estimates are used for evaluation. Equally important to the direct uses of improved cost predictors are the indirect uses. For example, predictors can be sought that relate requirements and resources to the methods used to control costs. With such predictors, one can compare alternative methods and staffing policies and select tools, techniques, and procedures that will tend to reduce costs.

Granted that cost estimation is important in programming, how is this activity now being performed? At the start of a project, when the user and the program developer have agreed upon the gross system requirements, the developer estimates the amount of work to be done based upon (a) the programs and procedures that have to be designed, implemented, tested and documented; (b) the analysis and experiments that may have to be conducted; and (c) new utility programs that may be needed. If possible, comparisons of the new system with existing systems are made in the hope of finding a cost-estimating guideline. A first estimate is made for the resources (men, machines, facilities and travel) required to do the work in the scheduled time. When these estimates are matched against their availability, the schedule may be adjusted accordingly. In addition, alternate proposals may be generated to reflect trade-offs between scheduled time, system requirements and costs. For a more detailed cost analysis, some prototype tasks may be completed and costed to determine the expected level of complexity and nature of problems.

An alternative and probably more frequent approach to costing is to estimate the number of program instructions using experience with similar programs as a basis. The number of instructions provides an intermediate parameter that is then converted to man months and computer hours by various rules of thumb. Man months and computer hours are then converted to dollars by multiplying by average expected rates. Finally, funds for supporting equipment, supplies, office facilities, travel, overhead and general administration are added to produce the total cost.

The current techniques for cost-estimating are not very accurate. Projects frequently require more resources than were originally estimated, even with ample safety factors introduced. Some reasons for this lack of success are the following:

a. Lack of agreement on terminology. The few "standards" that do exist do not contain a commonly accepted set of terms to describe the programming process, the programming products, and the personnel involved with these.

b. Poor definition of product quality. The lack of standard measures of product performance and product quality hampers comparison of costs among the various program systems. For example, the common use of a cost per instruction to compare programs does not recognize that radically different quantities of resources may be needed to develop two programs each of the same length should they differ in complexity, language used, programmer experience level, and the degree to which they were clearly specified at the start.

c. Poor quality of cost data. Present cost collection methods are not geared to accumulate data by product and by function to be performed. Therefore, costs that are collected by various organizations are difficult to compare.

d. Nonquantitative nature of many factors that contribute to cost. Programming costs are strongly influenced by many factors that are presently difficult to quantify such as the proficiency of the programming staff and the quality of management.

Despite these difficulties, this project was undertaken as a first step, in the hope that estimating the costs of programming products could be made a more systematic and reliable process.

Scope of this Project

The basic problem in cost estimation is: given the requirements for a computer program, what types and quantities of resources are needed to develop such a program? A further question is concerned with how these resources should be (or more realistically, could be) applied over time, i.e., the scheduling problem. One difficulty in cost estimation is that

the relationship between requirements and cost is not known. That is, the present ways of stating requirements cannot be readily interpreted in terms of the work to be done. Further, the resources, particualrly the programmers, cannot be characterized in such a way as to predict the work that they can do. A solution of the problem in the long run must involve finding ways to characterize work to be done, requirements, and resources, so that one can be translated into the others.

In this study we have tried to relate requirements, resources, and certain indicators of management practice to costs, using experience data and statistical techniques. To introduce some semblance of rigor, we defined a population (a) by limiting the scope of programming activities to program design, code and test and (b) by considering for purposes of comparison the concept of a "data point" defined as the smallest set of instructions

- whose purpose is defined by someone other than the programmer,
- which is deliverable to the user (customer) as a package, and
- which is loaded into the computer as a unit or system to achieve the stated purpose.

No attempt was made to further differentiate programming efforts. The many factors of programming language, programmer experience, and complexity, that may be used to explain differences in cost were identified as independent variables and tested for their significance by means of regression analysis.

We addressed the problem of estimation at the beginning of the program development effort and did not, therefore, include the costing of program changes. The problem of costing changes is one worthy of a thorough investigation. When changes are proposed, some experience has already been accumulated in the design of the program and relationships have been formed with the user, whereas at the beginning of the project the estimator has far less information. In estimating the cost of changes, one is concerned with both the additional instructions and documentation that have to be prepared and the "scrap" instructions that must be discarded. Also, in costing chnages, one must consider the effects of the change upon the entire program system. Therefore, the costing of changes will usually require more accuracy and probably include more detail of additional factors.

To conduct this analysis, we gathered data by questionnaire for twenty-seven completed programs. With these data, we developed analytical procedures using tested statistical techniques. Realizing that we were engaged in a search process and that our sample size was too small to achieve high confidence predictors, we aimed for two results:

a. A new questionnaire with improved ideas on data to be collected

b. Demonstration that the approach and methods used could lead to useful results

Our experience with respect to these objectives is discussed next.

III. APPROACH AND METHODS

Introduction

In this study, we used multivariate regression analysis as the basic analytical tool. The mathematical basis and theory of regression and correlation analysis will not be discussed in this report. Several good texts are listed among the references (1), (2) and (3). Regression analysis techniques have been used quite successfully to derive estimating relationships for determining the reliability of electronic equipment (4), the cost of overhauling ships (5), and the initial cost of tooling for aircraft production (6). To our knowledge, this study is the first application of such techniques to the problem of deriving cost-estimating relationships for the development of computer programs.

Since our statistical analysis and the associated work were exploratory, we felt it important to discuss the methods and techniques used and to review their relative success in some detail. We have included the problems and procedures of both the data collection and statistical analysis. This section is, therefore, concerned with methods only. Data and results of the analysis are discussed in the next section of the report.

Data Collection

Design of the Questionnaire. The organization of the questionnaire (see Appendix I) paralleled the organization of the cost factors discussed in Volume I of this study. Each of the six categories of factors comprised a section in the questionnaire. This organization permitted easy separation of the questionnaire, so that each section could be easily delegated to the people most qualified to complete it. The six parts of the questionnaire were:

1. Operational Requirements and Design
2. Program Design and Production
3. Data Processing Equipment
4. Programming Personnel
5. Management Procedures
6. Development Environment

The first two parts address the question, "What was the job to be done?" The next two ask, "What were the available resources?" and the last two ask, "What was the nature of the working environment?"

In the first volume of this series, we identified, organized, and discussed about 50 factors that were advanced as having an influence on the cost of computer program development. The first task in this project was to reformulate the factors so that they could be quantified in the program development efforts that were studied. This work was reflected in the questionnaire, the "instrument" used for data gathering, i.e., the presumed cost factors became items in the questionnaire, and later, variables in a statistical analysis.

The skillful design of the data questionnaire is a vital task in research of this kind, for it is on the basis of information obtained from this instrument that the validity of the approach rests. To construct sound questionnaires, certain basic principles of design must be adhered to. Three of the most useful principles are reliability, validity, and face validity. In designing the original questionnaire, reliability and validity were somewhat neglected, whereas face validity was emphasized.

Questionnaire reliability[1] can be viewed as the consistency with which a given pattern of responses is obtained from replication of the survey to identical or alternate respondents. We realized that poorly structured items present opportunities for ambiguous interpretations and inconsistent responses, and hence, lower the overall reliability of the instrument. Although, in this iteration, we treated each item as an independent variable in the analysis, we plan, in the next iteration, to explore aggregation techniques for grouping similar items into indices. This will tend to improve the reliability of questionnaire variables and preserve sources of cost variance that might otherwise be ignored.

The second principle, validity, concerns the extent to which the variables will predict costs. Statistical techniques are available, under appropriate circumstances, for testing, selecting and grouping items that will enhance overall questionnaire validity. Validity and reliability are interrelated in that the reliability of the questionnaire sets a limit on the validity it may achieve. Thus, increasing the reliability of the instrument will tend to increase its overall validity, provided the items remaining in the questionnaire retain their individual validities.

---

[1] Reliability, as used here, concerns the phenomena of errors or differences in measurement obtained when a characteristic of a given object is measured several times by instruments. This useful concept has been widely employed in psychological and educational fields. In the physical sciences, reliability of measurement is usually subsumed under the alternate topic, errors of observation.

The third principle, face validity, is essentially the meaningful quality that the questionnaire imparts to its respondents. Questionnaires having items with good face validity generally tend to create a favorable attitude among respondents, thereby increasing the likelihood of reliable responses, and consequently, allowing the inherent validity of the instrument to be achieved.

Of the characteristics mentioned above, the one on which the most emphasis was placed was the principle of face validity, i.e., meaningfulness and answerability of questions. To insure some degree of consistency in the understanding and answering of the questions, it was necessary to define terms within the body of the questionnaire. For example, such words as data base, instruction, parameter test, innovation, and many others do not enjoy a desirable degree of standardization and were, therefore, defined when used in a question. This technique was fairly successful and should be used even more extensively in the future.

In addition to face validity, we considered the accuracy of the data. To determine the accuracy of the responses of 44 key items of 93 in the questionnaire, we asked responders to assess the accuracy of their own answers to these items. They coded their assessment according to the following three categories:

Data Accuracy Index

Record	Memory	Judgment
1 Very accurate	4 Accurate recollection	7 Confident
2 Good estimate	5 Good guess	8 Good guess
3 Unreliable	6 Very hazy	9 Estimate

Appendix IV contains a frequency count of the estimated accuracy of the responses to each of the 44 questions. These responses were not used in any explicit way. If the resulting regression equations had displayed smaller confidence limits a closer examination of the accuracy of the input data would have been made to more completely insure our confidence in the results.

Design of the Sample. In the classical sense, there was no rigorous design of the sample. To expedite the analysis for this first iteration, only data within the system Development Corporation were collected. The types of programs for which data were collected, however, represented a fairly broad range: responses were received for operational programs, utility programs, and support programs, all within the category of command and control systems.

As pointed out earlier, two definitions were used to bound the data sample. First, the same set of programming activities comprised the program development effort in each observed case. We defined the scope of the programming job to begin with the program design activity and to end with program test (not including system test). Some questions were asked, however, about participation in the operational design activity. These activities of the programming process used as a base are described in Reference (7).

Second, a program unit, i.e., a "data point," a member of the data smaple, was defined to be: the smallest set of instructions (a) whose purpose is defined by someone other than the programmer, (b) which is delivered to the user or customer as a package, and (c) which is loaded into the computer as a program unit or system to achieve the stated purpose or objective. By this definition, a program data point can be an operational program, a utility program, or even an experimental or prototype program. The user of the program may be the buyer, or he may be another programmer, as in the case of a utility program.

Ideally, the number of data points for analysis should equal or exceed the total number of variables being considered for inclusion in the cost prediction equations. In addition, the points should range uniformly across the cost domain in which we wish to make estimates, i.e., the mathematical surfaces fitted by regression analysis should be securely anchored in the solution space and not subject to excessive translation or rotation when cross-validated to new data samples. A basic problem in this study was the small sample size. An excessive imbalance between number of data points and number of variables led to lack of complete confidence in rejecting potential predictor variables and contributed to the somewhat large confidence limits that characterize the prediction equations derived. Two associated problems created by survey limitations were (a) a poor distribution of program sizes measured in machine language instructions (i.e., many small programs, few large programs), and (b) a probable organizational bias in examining the experience of only one company. These problems will be discussed in more detail later in the report.

Administration of the Questionnaire. The questionnaire, instructions for its completion, and background information on the objectives of the project were sent to the managers of the three Divisions responsible for developing computer programs within the Corporation.[1] We suggested the major contract areas within these Divisions where we felt there would be a number of meaningful data points. We further suggested that the subordinate managers

---

[1] Air Defense Division
Washington Division
Command Control Division

responsible for the development of these programs determine how best to partition their program systems in accordance with our definition of a data point. Each questionnaire (i.e., data point) was then further delegated to the people most qualified to provide the required information. As mentioned above, the questionnaire was designed to be easily divided and delegated.

We held short meetings with the recipients of the questionnaires to explain the intent of the questionnaire and to answer questions regarding the information requested. Efforts on the part of the responders in completing the questionnaires ranged from one to five man days. After receipt of the completed questionnaires, we effected follow-up communications where necessary to request explanations of answers that were unclear, ambiguous or nonresponsive.

Statistical Analysis

General Approach. The basic statistical technique used was multivariate regression analysis. Mathematically, this procedure involves the derivation of the equation of a surface that fits as closely as possible the observed data points (see References 1, 2, and 3). In using statistical techniques to solve a heretofore completely unstructured problem, we were faced with three major problems: (a) the recognized unreliability of the data, (b) the relative scarcity and poor distribution of data points in the sample, and (c) the unfavorable ratio of data points (sample size) to variables, i.e., many more variables than available data points. Despite these problems, the statistical techniques employed were sufficiently robust[1] to produce meaningful results.

During the time allotted for this study, little could be done to solve the first two problems. The basic methods of regression analysis and factor analysis were supplemented by correlation analysis and intuitive analysis in order to deal with the problem of imbalance between data points and variables. Initially, the analysis of cost factors in computer program development led to the identification of 93 variables (i.e., questionnaire items) that were believed to be associated with costs. Generally speaking, the number of data points should have exceeded the number of such variables to obtain a trustworthy analysis. Thus, in this problem, we would have preferred several hundred data points to use as a basis for selecting the best variables and determining their proportionate relevance in cost estimation. As it turned out, a major analytical problem concerned the reduction of the total number of potential predictor variables to a lesser

---

[1] A robust technique is considered here to be one that is relatively insensitive to departures from the assumptions and conditions on which it has been theoretically based.

number of representative variables while proceeding to the initial development of prediction equations. Even with the unfavorable data point-to-variable ratio, it was possible to apply statistical techniques as an aid in selecting desirable variables. However, to compensate for the inherent instability of statistical procedures based on small and poorly distributed samples, we relied heavily upon the program system development knowledge and experience available to us.

Other approaches, consistent with the fundamental goals of multivariate regression analysis, were used to select variables for further analysis. In general terms, the criteria for selection of the "best" variables were as follows:

1. Validity--the extent to which each predictor variable individually accounted for cost variance. Initially, the correlation coefficients of all variables with respect to major costs were examined. As analysis progressed, standardized regression coefficients on specific costs were used to refine the selection.

2. Independence--the extent to which each predictor variable was free of relationship to other predictor variables. This was observed by examining the intercorrelations among predictor variables.

3. Confidence--the extent to which each predictor variable, when included in a multivariate prediction equation, would tend to increase the confidence that can be placed in the predicted cost parameter. The available theory provided useful indices such as standard errors of estimate and confidence limits. Confidence estimation was the key aspect of the current analysis. This important topic is discussed more fully in the section describing the prediction model.

4. Distribution Quality--the extent to which each predictor variable tended to be distributed without large gaps and without severe skewness to either high or low values. On occasion, transformations of variables by logarithms were employed.

5. Missing Data--the extent to which each predictor variable was free of missing or approximated data. A working principle suggested that variables that were so difficult to assess as to have frequent missing values were probably poor variables for practical prediction purposes.

6. Intuitive Considerations--general opinions and experience concerning the usefulness of a variable for prediction purposes.

While intuitive considerations pervaded the entire variable selection procedure, considerations of validity, independence, and confidence were weighted most heavily in the regression analyses, and considerations of distribution characteristics and missing data were primarily confined to

the initial analysis of raw data. However, a few appealing variables having inferior distribution characteristics or approximated data were included in some of the regression analyses but were given a low selection priority.

Prediction Model. The statistical analyses for this study were based on the foundations of multivariate regression analysis. Although this statistical approach attempts to provide a model in which the cost of computer programs is related structurally to independent factors, the major emphasis in the model used here is not on the statistical rigor with which the prediction equations are derived but on the practical accuracy and usefulness of the equations in the actual task of estimating computer programming costs. Simple predictive efficiency, although acknowledged, is not emphasized. Instead, the goal is to provide a tool that is sufficiently valid to be useful outside of the particular data pattern on which the empirical analysis is based.

Statistical tests available for evaluating the estimating efficiency of equations from their sample data are important but insufficient indicators of the quality of a model of this type. Experience has frequently revealed that equations, although satisfying rigorous estimation criteria in the sample from which they were derived, still perform rather poorly when applied to new data. The ultimate value of a prediction equation lies in the extent to which it can make useful predictions outside of the data sample on which it was based. This, of course, places a great responsibility on the research program in acquiring data sufficient in quantity, representativeness and practicality to warrant application to the domain in which predictions are to be made.

Initially, it was assumed that an enduring linear relationship exists between costs (Y_k) and various suitably weighted subsets of predictor variables X_a, $X_b, \ldots X_m$ Mathematically, the basic task of analysis involved the fitting, by least-squares procedures, of hyperplanes (i.e., flat surfaces in three or more dimensions) to a sample of data points arrayed in m + 1 orthogonal dimensions. This model may be compactly expressed as follows:

$$Y_k = A_k + \sum_{i=1}^{m} B_i X_i + E_k \qquad (1)$$

where: Y_k is the value of the kth cost dimension to be estimated.

A_k is a constant that may be either positive or negative in all estimates for a particular Y_k.

B_i is the weight to be assigned to the ith predictor variable to optimize the overall accuracy of the equation.

X_i is the numerical value for the ith predictor variable.

m is the number of predictor variables used in the prediction of Y_k.

E_k is the portion of Y_k that cannot be estimated by any weighting of the X_i. This is known as the error term. It may be positive or negative and will vary randomly from data point to data point.

Although the linear prediction model can be extended to the quadratic case, it was not used in this study due to the relative scarcity of data points. However, future analysis may suggest this type of modification.

In multivariate estimation, the critical element in the equation is the E_k term, because this defines the statistical confidence that one may place in the equation. At one extreme, the E_k term may be zero for all observations, which would define a perfect estimating equation. At the other extreme, the contribution of the X_i would be zero, and the equation would be worthless. In this case, the distribution of E_k would be approximated by the standard deviation of the Y_k values from the arithmetic mean of Y_k. The mean would, in all such cases, be the only reasonable estimate for any Y_k because it would lead to the least error of estimate, overall.

In actual practice, the distribution of E_k values will lie somewhere between the two extremes described above. For this purpose, a fundamental statistical parameter called the standard error of prediction is available. This device was designed to be used when the estimation errors are expected to be approximately normally and independently distributed. The formula for this parameter is as follows:

$$\sigma(Y_k) = \sigma_E \sqrt{1 + 1/N + \sum_{i,j=1}^{m} c_{ij}\, x_i\, x_j} \qquad (2)$$

where: $\sigma(Y_k)$ is the standard error of prediction for an individual Y_k derived from selected X_i. This parameter defines the limits within which one can expect the true Y_k to fall two-thirds of the time.

σ_E is the standard error of estimate, defined as the root mean square error adjusted for sampling bias and the number of predictors used, i.e., $\sqrt{\frac{\Sigma E^2}{N-m-1}}$, where E = actual Y minus Y computed from the regression formula.

N is the number of data points on which the estimation weights are based.

m is the number of predictor variables.

i, j are subscripts used to define cross-multiplication among predictors.

c_{ij} are multipliers used to weight the cross-products of predictor deviations from the mean. These multipliers are obtained from the inverse of the augmented correlation matrix by the following formulas:

$$c_{ii} = \frac{a_{yy}\, a_{ii} - a^2_{yi}}{(N-1)\, \sigma^2_i\, a_{yy}} \tag{3}$$

and

$$c_{ij} = \frac{a_{yy}\, a_{ij} - a_{yi}\, a_{yj}}{(N-1)\, \sigma_i\, \sigma_j\, a_{yy}} \tag{4}$$

where: a_{yy} is the value of the inverse element for the dimension to be predicted.

a_{ii} is the value of the diagonal inverse element for the ith variable.

a_{yi} is the value of the inverse element at the row-column juncture of y and i.

a_{yj} is the value of the inverse element at the row-column juncture of y and j.

N is the number of data points.

σ_i and σ_j are unbiased estimates of the population standard deviation for the predictor variables arrayed in i rows and j columns.

x_i, x_j are the deviations of the predictor (X_i) values from their respective means.

In the particular case where all predictor variables are taken at their respective arithmetic means, the above formula for the standard error of prediction reduces to:

$$\sigma(Y_k) = \sigma_E \sqrt{1 + 1/N} \tag{5}$$

For example, when $N = 26$, $\sigma(Y_k) = 1.02\, \sigma_E$

It is customary in confidence estimation to use approximately $\pm 2\sigma(Y_k)$ to establish the 95 percent confidence limits for a predicted value. This provides the extremes within which the true Y_k value can be expected to

fall 95 percent of the time. For analyses based on small samples, the confidence limits must be expanded to account for the lesser stability of the predictions. For example, when the number of data points is 26 and the number of predictor variables is 4, one must use $\pm 2.08\sigma(Y_k)$ rather than $2\sigma(Y_k)$ to establish the 95 percent confidence limits. The use of these devices provides a safeguard against unwarranted acceptance of statistical results derived from small samples. The results of using equation (5) for determining confidence limits are included in the tables of Appendix VII, which summarize the results of the correlation and regression analysis. In subsequent research, it is anticipated that the more complete calculation of confidence limits using equation (2) will be appropriate. This technique will allow the calculation of confidence limits for specific values of the predictor variables, in each use of an estimating equation.

Selection of Predictor Variables. As noted above, a primary problem facing the investigators was to reduce the number of predictor variables to be submitted to the regression analysis. Clearly, the 93 predictor variables had to be reduced to less than 27 (the available number of data points) before the regression technique could be applied. In Appendix II, we have listed definitions of all predictor variables and their coding. These variables are, in actuality, the questionnaire items described in Appendix I. Appendix V is a validity table summarizing these same variables and their individual correlations with costs. Below we describe how we used the principles mentioned earlier to select or reject predictor variables for further analysis. These principles were applied in several overlapping phases: examination of raw data, correlation analysis, regression analysis and factor analysis. The results of the selection process are recorded in Section IV, Summary of Results.

1. Examination of Raw Data. The responses to the questionnaire were tabulated in a data matrix (Appendix III) in which each column (variable) was carefully examined. Ten of the original variables were immediately rejected for one or more of the following reasons:

   a. Lack of variance or a predominance of constant values. For example, if a yes or no question exhibited more than twenty identical responses, the variable was rejected.

   b. Identity with other variables. In cases where columns displayed identical or near-identical entries to other columns, a rejection of one of the variables was made.

   c. Poor distribution characteristics. If examination of the data revealed large gaps (discontinuities) or highly skewed (unbalanced) results, the variable was rejected.

d. Excessive amount of missing data. In cases where only a few cells were missing, the investigators filled these in as accurately as possible. However, if many entries were missing, predictor variables were rejected, but cost variables were retained for further analysis.

e. Apparently ambiguous question. If the majority of responses appeared to be incorrect, that is, not responsive to the intent of the question, the variable was rejected.

f. Dependence on other variables. For example, in a series of ratio or percentage variables adding up to 100 percent, one variable could be rejected as being dependent on the others.

g. Lack of strong intuitive appeal. This criterion generally pervaded the rejection of variables throughout the research.

2. Examination of Correlations. At this point, there were 83 "independent" and 15 dependent variables under consideration. The first computer run consisted of the computation of a 98 by 98 correlation matrix, which depicted the statistical relationship of every variable with every other variable. Each predictor variable was examined first for its correlation with costs as a preliminary means of checking its validity. Variables with low correlations and spuriously signed correlations were then considered for possible rejection. Because a considerable number of variables had to be rejected before regression analysis could be attempted, variables with low validity coefficients were not accepted unless they had strong intuitive appeal. In all cases where variables were selected or rejected, they were checked for meaningfulness, unambiguousness, availability, and general appeal. These criteria are, of course, all subject to the investigators' judgment and intuition. Highly valid predictor variables were examined for their intercorrelations. We realized that highly intercorrelated predictor variables, even though valid,[1] would

[1]Given approximately the same validity level among predictors, an equation based on more unique independent variables will be more trustworthy than one based on highly correlated variables; this is because multicollinearity increases the sensitivity of parameter estimates to such things as changes in the set of independent variables used, the relative presence or absence of extreme observations and the direction of minimization. This thereby reduces one's confidence in the usefulness of whatever structural estimates happen to emerge. Reference (8) provides a more thorough technical discussion of this important topic. Although no standard test of significance exists for evaluating the extent to which multicollinearity affects an equation, Formula (2) (standard error of prediction) is considered to be useful for evaluating competitive equations, since it takes into consideration the nature of the inverse, and consequently, the relative value of the determinants of the predictor intercorrelation matrices from which the equations have been derived. We were unable to utilize the standard error of prediction as an evaluation device in the current analysis due to lack of time and a suitable computer program.

tend to complicate the regression analyses should they be allowed to compete with each other in accounting for variance; to avoid spurious results such as negative regression coefficients for variables which are positively related to costs, we decreased the number of highly intercorrelated predictors by selecting the most appealing of the competing variables. This technique was also designed to increase the true independence or uniqueness of the predictor variables being considered for inclusion in the equations.

3. Examination by Regression Analysis. At this point, the number of variables selected for further analysis was over 50. This still greatly exceeded the number of available data points. Therefore, the number of variables was further reduced and divided into two groups. One group was labeled "most preferred" and consisted of 15 predictor variables; the other was labeled "satisfactory" and consisted of 21 predictor variables. At this point, multivariate analysis was introduced to further reduce the number of variables.

A multiple regression analysis program (9) along with an IBM 7094 computer were the primary computational tools used by the investigators, although other computer programs were also used in support of this effort. The linear multiple regression program we used can perform a complete analysis on as many as 80 variables, provided enough data points are available to justify the computations. The following quantities are computed and output: sums and sums of squares, means, sample size, standard deviations, the intercorrelation matrix, standardized and weighted regression coefficients, the standard error of estimate, the coefficient of determination, the multiple correlation coefficient, and the constant in the regression equation.

The program also selects subsets of independent variables that yield near-maximum multiple correlations (i.e., near-minimum residuals). Once the program selects a subset of, say, m variables, it computes and outputs the following statistics: values of a gradient selection index for each variable, standardized and weighted regression coefficients, the regression constant, the coefficient of determination, the multiple correlation coefficient, the shrunken multiple correlation coefficient, the standard error of estimate, the increase in the multiple correlation from the previous subset, the change in the shrunken multiple, the decrease in the multiple correlation from the complete set of independent variables and the corresponding F ratio. The program will continue selecting larger and larger subsets of predictor variables until a predesignated stop criterion is satisfied.

In the first regression analysis, we planned to use the subsetting feature and the computation of the standard error of estimate to assist in further rejecting variables. Specifically, we expected the minimum standard error of estimate to occur after the selection of about four to eight variables,

whereupon the remaining variables would be rejected. When first used, this technique did not give a clear indication of which variables to consider for further selection. In fact, the minimum standard error occurred in most cases after selecting all the variables submitted to it. We found that data point 5, an unusually large deviate in the multiple solution space, was the anomaly in the analytical process. When this point was removed on the fourth regression analysis, the computed solutions proceeded in a straightforward manner and subsets of variables with minimum standard error were readily apparent. Hence, the majority of results that follow are based on a sample of 26 data points rather than the 27 for which data were collected.

In the second regression analysis, the "most preferred" and the "satisfactory" variable (see Tables I and II in Section IV) groups were reexamined on the basis of all criteria, with special emphasis placed on identifying and rejecting redundant variables. The two groups were then consolidated into one group of 17 "best" variables that was subjected to further regression analysis. To continue the process of rejecting variables, we used standardized partial regression coefficients as a means for evaluation. Variables with coefficients less than .10 and those with an algebraic sign inconsistent with good judgment were generally rejected. The number of predictor variables associated with each cost variable was thus reduced to less than ten. These variables were then submitted to a third and fourth regression analysis, the results of which are described in Section IV.

We conducted a fifth regression analysis to completely eliminate the potential bias introduced by data points 4, 5, and 6, the extremely large programs, in terms of man months and number of instructions. This final analysis also used the number of delivered instructions as a predictor variable in place of the companion variable, number of instructions originally estimated. When all extreme data points were removed, the scatter plot relationships between costs and delivered instructions (see Figure 5, Section IV) were more meaningful and trustworthy than those using the after-the-fact reports of estimated instructions. Since both variables were collected simultaneously, this appeared to be a reasonable choice of alternatives.

Because the number of delivered instructions played such a dominant role in this study, a companion analysis was performed to derive an equation for estimating delivered instructions from other predictor variables, completely excluding the variable, estimated instructions. The results of this analysis are described in Section IV.

The following is a summary of the five regression analyses:

Analysis	Variables Considered	Comments
First	Tables I & II	We intended to reject variables appearing in results after minimum standard error of estimate was achieved (N = 27).
Second	Table III	We selected variables for further analysis on basis of satisfactory standardized regression coefficients and meaningfulness (N = 27).
Third	Table IV	Specific predictor variables were grouped with specific cost variables (N = 27).
Fourth	Table IV	We repeated the previous analysis with omission of data point 5 and also conducted a special analysis to derive an equation for estimating delivered instructions from other predictors (N = 26).
Fifth	Table V	A final analysis only on variable 84 (man months) with omission of data points 4, 5, and 6 (N = 24).

4. Examination by Factor Analysis. In addition to the techniques described earlier, we also initiated the use of factor analysis (10) as a means for studying the relationships among the cost predictor variables. This technique allowed the predictor intercorrelation matrix to be described by a smaller number of independent entities, called factors, that helped to account for the observed intercorrelations in the matrix. Using an IBM 7094 computer program (11), we obtained a table of factor loadings showing the relationship between each variable and each factor. Viewed geometrically, these loadings represent the projections of the variables (as vectors) on referent axes in an orthogonal multidimensional coordinate system. Since the referent axes are rather arbitrarily defined in the basic calculation process, they may be rotated to any position that will enhance the description of the original data. Another IBM 7094 computer program (12), employing a varimax method of factor rotation, was used in this study to achieve factorial description of the 83 variables in the predictor pool. The table shown in Appendix VI illustrates the results of using this approach.

   Factor analysis, like regression analysis, requires, among other things, a favorable data point-to-variable ratio for its successful application. Since the results shown in Appendix VI were based on only 26 data points drawn exclusively from one organization's experience, and the questionnaire used to obtain these points is in its first experimental phase,

they are not to be interpreted as definitive and exhaustive of the computer programming domain. Although the purpose of factor analysis is aimed more at description than at prediction, it was felt that this approach could provide a valuable adjunct to regression analysis in the search for unique and valid variables for predicting computer programming costs. Accordingly, in this study, the factor composition of variables was taken into consideration, along with the other criteria, when variables for regression analysis were selected.

Evaluation of Approach

As an initial exercise in analysis of programming costs, this study has outlined problem areas and suggested ways to continue both the data collection and statistical analysis more effectively. Most importantly, a revision of the questionnaire is indicated to improve the relevancy and clarity of the data to be collected. The analytical techniques just described, although powerful, appropriate tools for the examination of a highly complex multivariate problem, require a relatively large data sample to produce reliable and valid results. Since we did not have a large sample size, the results in the next section should be considered as examples of the anlytical techniques rather than recommended prediction devices. An improved questionnaire design that is pointed at minimizing the effort required to complete it probably will help us collect data from a larger and more representative audience.

Improvements of the questionnaire and data collection should focus on the following:

a. Improved definitions of terms. For example, terms such as data point, programming tools, concurrence, as well as many others are in need of more concise and explicit definition. This is especially necessary to collect meaningful, comparable data from organizations outside of SDC.

b. Design of dichotomous questions for ease of aggregation.[1]

c. Extension of the scope of the program development effort being examined to include system analysis, as well as installation and maintenance activities. Although difficulties may be encountered in analyzing a nonhomogeneous population, this larger view is much more realistic and logical in attempting to account for all the factors that affect the cost of programming.

---

[1] In the present analysis, the dichotomous variables fared rather poorly in predictor variable selection. It is quite probable that the small variance of such variables acted as a deterrent against their selection when they were matched against quantitative variables of much larger variance. Appropriate aggregation would allow higher variance with the resultant possibility that they, as a group, might better complement the quantitative variables and contribute to additional prediction accuracy.

d. Elimination of questions that produce response variables having a marginal or spurious contribution to costs. It would be highly desirable to reduce the size of the questionnaire by eliminating such items. However, this can now be done only with very low confidence with the sample data available.

e. Addition of questions to focus more on the actual data processing to be performed, the organization for program development and the relative value of the resulting program and its documentation.

f. More detailed and complete validation of the cost data to insure some degree of accuracy.

The success of multivariate analysis for cost prediction depends to a great degree on the clear and meaningful definition of variables and the ability to collect sufficient amounts of reliable data associated with these variables. Because the ultimate significance of specific variables, i.e., presumed cost factors, is unknown and very little data collection has been accomplished, the entire data collection and analysis process must be iterative. One objective of an analysis of past program development efforts is to establish a data collection and reporting plan for new development efforts. Descriptive terms must be challenged and often redefined and new terms and definitions created as work progresses. Research data collection and processing procedures must also be challenged, evaluated, and perhaps modified. As more and more relevant data become available, the output of a research program of this type can be expected to become more and more accurate and valuable. The maximum value of this kind of analysis can be obtained by submitting results to managers for actual use. Finally, the ongoing nature of the data collection program suggested above will allow the timely assessment of important new factors such as advanced programming techniques, equipment and procedures that are being introduced into computer program development.

IV. SUMMARY OF RESULTS

Introduction

This section details the results of the predictor variable selection process described earlier, presents some selected regression equations derived from the statistical analysis, and interprets the results in terms of their validity, usefulness, and implications for further work. Associated with each regression equation are error indices (residuals) that reveal the specific portion of the cost variance unaccounted for by the equation. When plotted graphically, these residuals readily describe how the estimated or computed value of the cost compares with the actual value. For purposes of illustration, this section presents the results of the regression analyses for three cost variables: (a) man months for program design, code and test; (b) computer hours; and (c) number of delivered instructions. Data plots for these cost variables are also provided.

A detailed summary of the correlation and regression analyses for all cost variables is presented, in tabular form, in Appendix VII. These tables indicate the variables considered in the final analysis and present such pertinent statistics as the means, standard deviations, validity coefficients, intercorrelations, standardized regression coefficients and confidence limits.

Discussion

The primary objective of the analysis described in the previous section was the development of reliable cost-estimating equations. Assuming an adequately large and representative sample, these equations will predict cost such that the probable errors of prediction will be minimized. Not only is the dependent variable (cost) of great interest to the user of such equations, but the regression coefficients themselves imply relative significance concerning the independent (predictor) variables comprising the estimating equation. However, the reader should not assume that control of statistically derived predictor variables will necessarily control costs. The significance is primarily statistical and not necessarily causal. The degree of causality is related to such things as the meaningfulness of the selected variables and the relative presence or absence of program quality and performance considerations in cost estimation. For example, if in the equation (see Figure 1) for the cost variable, man months, we reduce the numerical value for the predictor variable, number of external documents, we then reduce cost, it is also possible that the quality of the program may be reduced drastically. The equations in this document are primarily illustrative of the research methodology and are not recommended for use in actual program development efforts. On the other hand, we encourage the use of these equations on an experimental basis, e.g., to supplement and compare with other estimation techniques. Further, reports of such usage will be extremely valuable in our continuing research.

The relatively poor distribution of data in the cost domain requires some discussion. The 27 data points collected in this study consisted of 3 extremely large programs, 3 moderately large programs, and 21 relatively small programs in terms of number of instructions. Mathematically, this involved the fitting of a regression surface across large areas of solution space where no data points were observed. As a result, the equation of the cost surface favored the larger, more expensive programs represented by a small percentage of data points. In fact, the three largest points affected the investigation so adversely that they were all purged in the final regression analysis. During the analysis we began the purging by dropping data point 5, the single largest data point, so that the bulk of the results reflect the analysis of 26 data points. A final regression analysis for cost variable 84 (man months) only was based on 24 data points (the three largest data points: 4, 5 and 6 removed).

One goal in our selection of predictor variables was to use those that would be available or easily estimated at the beginning of a programming effort. In some cases, in the resulting equations, the estimation of the predictor vairables is easy; in others, a new problem arises. The best example of this is the variable, number of computer program instructions. This variable has significant correlation with cost; however, managers historically have had a difficult time in estimating instructions. In the current sample, a high correlation (.94) was observed between estimated and delivered instructions.[1] Since data on both variables were collected simultaneously, we suspect that some contamination may have occurred to yield this high correlation. Our approach to this situation was to use the variable called delivered instructions as a key predictor (which, incidentally, increased the confidence in the man months equation by 60 percent) and then to perform a separate regression analysis to predict delivered instructions without using estimated instructions as a variable. In general, this approach involves reducing the larger problem to cost estimation to a series of smaller and, hopefully, less complex problems of estimating the components of cost.

The following section outlines the sequence of steps we used in selecting variables for regression equations.

Predictor Selection

In the section on methods, we pointed out the need to reduce the number of predictor variables before a meaningful regression analysis could be attempted. A principal characteristic of regression analysis is that, as the number of potential predictors increases to approach the number of data points, the solutions (i.e., regression coefficients) tend to be spurious. This fact viatiated computerized variable selection capability, which is dependent, in large part, on the computation of reliable standardized regression coefficients. Before the variable selection capability of regression analysis could be used with some degree of legitimacy, the original set of potential predictors had to be reduced by correlation analysis, intuitive analysis, and factor analysis. Part of the total correlation matrix (a validity table), i.e., the relationship of each predictor variable to each cost variable, is presented in Appendix V. The remainder of the matrix, the intercorrelations of all the predictor variables, has been withheld to conserve space.

---

[1] It should be pointed out that estimated instructions was originally considered a predictor variable and delivered instructions a cost variable.

The first predictor variables selected for regression analysis are shown in Tables I and II. These were chosen on the basis of the following criteria: high validity, uniqueness, meaningfulness, availability, and general appeal. Except for a few cases, the variables omitted from Tables I and II were no longer considered in the analysis. Table I contains a list of the 15 "most preferred" variables, indicating their correlation with man months, the variable number, and a comment that further characterizes them. Table II contains the selection of an additional 21 "satisfactory" variables with similar descriptive information. With the small sample size available, the probability of making unwarranted rejections of variables by the methods used is high. The two separate regression analyses performed on the variables in Tables I and II were followed by a further selection of variables, results of which are shown in Table III. In general, the variables listed in Table III were selected because they ranked high in validity and meaningfulness.

While all the potential predictor variables in the first and second regression analyses were regressed against fifteen cost variables (84 through 98), in the third regression analysis we selected specific groups of predictor variables to be regressed against eight major cost variables on the basis of previously computed satisfactory standard regression coefficients and meaningfulness. The results of these selections are shown in Table IV. The remaining cost variables were either eliminated from further analysis or combined into new dependent variables.[1] All the variables in Table IV were run again in a fourth analysis using 26 points, data point 5 having been omitted. In the fifth regression analysis, data points 4, 5, and 6 were eliminated and the correlation analysis was repeated to select variables on the basis of new correlation coefficients (see validity table, N = 24, Appendix V). Table V lists the predictor variables considered in this regression analysis, which was completed only for cost variable 84 (man months).

---

[1] Variables 86 (average number of programmers), 92 (computer hours for program design change), 93 (pages of documents for program design change) and 97 (number of other personnel) were considered to be poorly conceived and of doubtful value, while variable 99 (total man months) became the sum of variables 84, 85, 89, and 98; variable 100 (man months for program design change) became the sum of variables 91 and 94. All variables are further defined in Appendix II.

TABLE I. FIRST REGRESSION ANALYSIS

Most Preferred Variables

Variable No.	Correlation* with Man Months (84)	Short Variable Description	Comments
11	.89	Number of instructions in original estimate (1000's)	Dominant predictor
18	.83	Number of input message types	Intercorrelated with 11, estimated instructions
21	.80	Number of subprograms	Intercorrelated with 11, estimated instructions
39	.78	Number of external document types	Intercorrelated with 11, estimated instructions
17	.70	Number of data base classes ($\log_{10}$)	Intercorrelated with 18, input messages, and 16, words in data base
33	.56	Number of programming tools	Intercorrelated with 31, time of peak program changes
38	.45	Number of internal document types	Intercorrelated with 11, estimated instructions
44	.41	Number of words in core storage (1000's)	Intercorrelated with 64, terminations per month
26	.36	Percentage of decision-making instructions	High appeal
76	.30	Number of agencies required for concurrence	Intercorrelated with 77 and 78, experience and decision capability of agencies
32	-.30	Language type used	Possibly spurious algebraic sign
23	-.29	Percentage of clerical instructions	Low appeal, meaningful sign
8	.22	Number of commands	High appeal, low validity
29	.20	Timing constraint	High appeal, low validity
5	-.12	How well operational requirements known	Meaningful sign, low validity

*These coefficients, based on 26 data points, changed significantly when all the extremely large data points were removed. See the Validity Table for N=24 in Appendix V.

TABLE II. FIRST REGRESSION ANALYSIS

Satisfactory Variables

Variable No.	Correlation* with Man Months (84)	Short Variable Description	Comments
83	.92	Number of trips x average miles/trip	Highly correlated with 11. estimated instructions
30	.78	Number of program design changes	Very difficult to estimate
13	.69	Number words in tables and constants	Not available early in development
10	.67	Complexity rating	Needs more quantitative definition
65	.46	Number of hires per month	Moderate validity
64	.43	Number of terminations per month	Moderate validity, meaningful sign
42	-.43	Computer operation adequately documented	Meaningful negative sign
28	.40	Program design constraints: insufficient memory	Moderate validity
41	-.32	Was computer time adequate for parameter test	Meaningful negative sign
12	.23	Ratio: new instructions/ delivered instructions	May be difficult to estimate
1	.17	Innovation in operational system	Low validity, high appeal
72	-.17	Document for cost control	Low validity, meaningful sign
81	.17	Program developed at site different than operational	Low validity, high appeal
60	.08	Ratio: operational design programmers/total programmers	Low validity, high appeal
80	.07	Computer operated by another agency	Low validity, high appeal
56-58	.41,59,.03	Index of experience for Types I, II, and III	
52-54	.40,.13,-.34	Percent of Programmers by Types I, II, and III	

*These coefficients, based on 26 data points, changed significantly when all the extremely large data points were removed. See the Validity Table for N=24 in Appendix V.

TABLE III

SECOND REGRESSION ANALYSIS

Variable No.	Correlation* with Man Months (84)	Short Variable Description	Comments
11	.89	Number of instructions in original estimate (1000's)	Variable 16, number of words in data base, was brought into the list because it was statistically more compatible with 11, number of instructions in original estimate, and other prominent predictors than was 17, number of D/B classes. Variables 46, number of displays, and 69, plan for unavailable computer, were also re-entered due to their relative uniqueness and high appeal. However, both were later rejected for reasons of low predictive contribution.
21	.80	Number of subprograms	
39	.78	Number of external document types	
13	.69	Number words in tables and constants	
10	.67	Complexity rating	
16	.65	Number of words in data base ($\log_{10}$)	
38	.45	Number of internal document types	
64	.43	Number of terminations per month	
44	.41	Number of words in core storage (1000's)	
26	.36	Percentage of decision-making instructions	
23	-.29	Percentage of clerical instructions	
8	.22	Number of commands	
46	.22	Number of displays	
72	-.17	Document for cost control	
69	.15	Plan in the event of unavailable computer	
5	-.12	How well operational requirements known	

*These coefficients, based on 26 data points, changed significantly when all the extremely large data points were removed. See the Validity Table for N=24 in Appendix V.

TABLE IV

THIRD REGRESSION ANALYSIS (N=27)
FOURTH REGRESSION ANALYSIS (N=26)

Cost Variables	(84) Man Months Prog. Des., Code, Test	(87) Months Elapsed	(88) Computer Hours	(90) Delivered Instructions	(90) Delivered Instructions*	(96) Pages External Documents	(99) Total Man Months	(100) Man Months Prog. Des. Change
Predictor Variables	(11) Estimated instructions	(11) Estimated instructions	(11) Estimated instructions	(11) Estimated instructions	(5) Operations requirements known	(11) Estimated instructions	(11) Estimated instructions	(11) Estimated instructions
	(26) % Decision making instructions	(26) % Decision making instructions	(10) Complexity	(8) No. of commands	(13) Words in tables	(18) No. input messages	(39) No. external documents	(23) % Clerical instructions
	(39) No. external documents	(39) No. external documents	(26) % Decision making instructions	(38) No. internal documents	(16) Words in data base	(8) No. of commands	(10) Complexity	(38) No. internal documents
	(10) Complexity	(64) No. terminations per month	(16) Words in data base	(44) Core size	(18) No. input messages	(5) Operations requirements known	(26) % Decision making instructions	(26) % Decision making instructions
	(38) No. internal documents	(44) Core size	(38) No. internal documents	(72) Plan for cost control	(21) No. subprograms	(72) Plan for cost control	(38) No. internal documents	(10) Complexity
	(16) Words in data base	(16) Words in data base	(64) No. terminations per month	(18) No. input messages	(44) Core size	(39) No. external documents	(16) Words in data base	(18) No. input messages
	(64) No. terminations per month	(13) Words in tables					(64) No. terminations per month	(13) Words in tables
		(10) Complexity						(8) No. commands

*Without using variable 11, estimated instructions.

Cost-Estimating Equations

Four of the resulting cost-estimating equations from the fourth and fifth regression analyses are presented here for illustrative purposes, while eight equations of interest are presented with additional statistical detail in Appendix VII. The first equation of interest, based on a sample of 26 data points, estimates man months for program design, code, and test:

$$Y_{84} = 2.7X_{11} + 121X_{10} + 26X_{39} + 12X_{38} + 22X_{16} - 497$$

Standard error of estimate = 138 M/M

95% confidence limit at the mean = $\pm$295 M/M

Variables

84	Man months for program design, code, and test
11	Number of instructions in original estimate (in thousands)
10	Complexity rating (scale 1-5)
39	Number of external document types
38	Number of internal document types
16	Number of words in data base ($\log_{10}$)

Figure 1, a plot of actual cost versus costs estimated with this equation, shows residuals (estimating errors) as deviations from a 45-degree line. Table 1 of Appendix VII describes the statistical characteristics of the variables used in this equation.

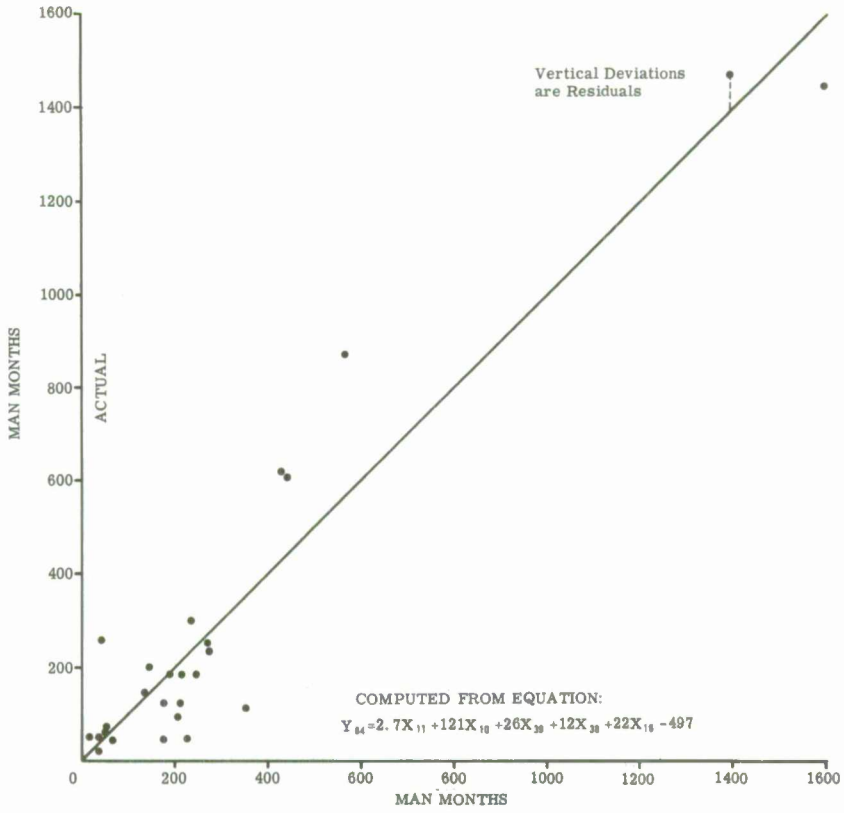


Figure 1. Actuals vs Computed for Cost Variable 84--Man Months for Program Design, Code and Test (N = 26)

The second equation estimates computer hours and is also based on the same 26 data points:

$$Y_{88} = 21.5X_{11} + 985X_{10} + 197X_{16} - 3468$$

Standard error of estimate = 905 hours

95% confidence limit at the mean = $\pm$1911 hours

Variables

88 Computer hours

11 Number of instructions in original estimate (in thousands)

10 Complexity rating (scale 1-5)

16 Number of words in data base ($\log_{10}$)

Figure 2 is a comparison of actuals versus computed values for this equation.

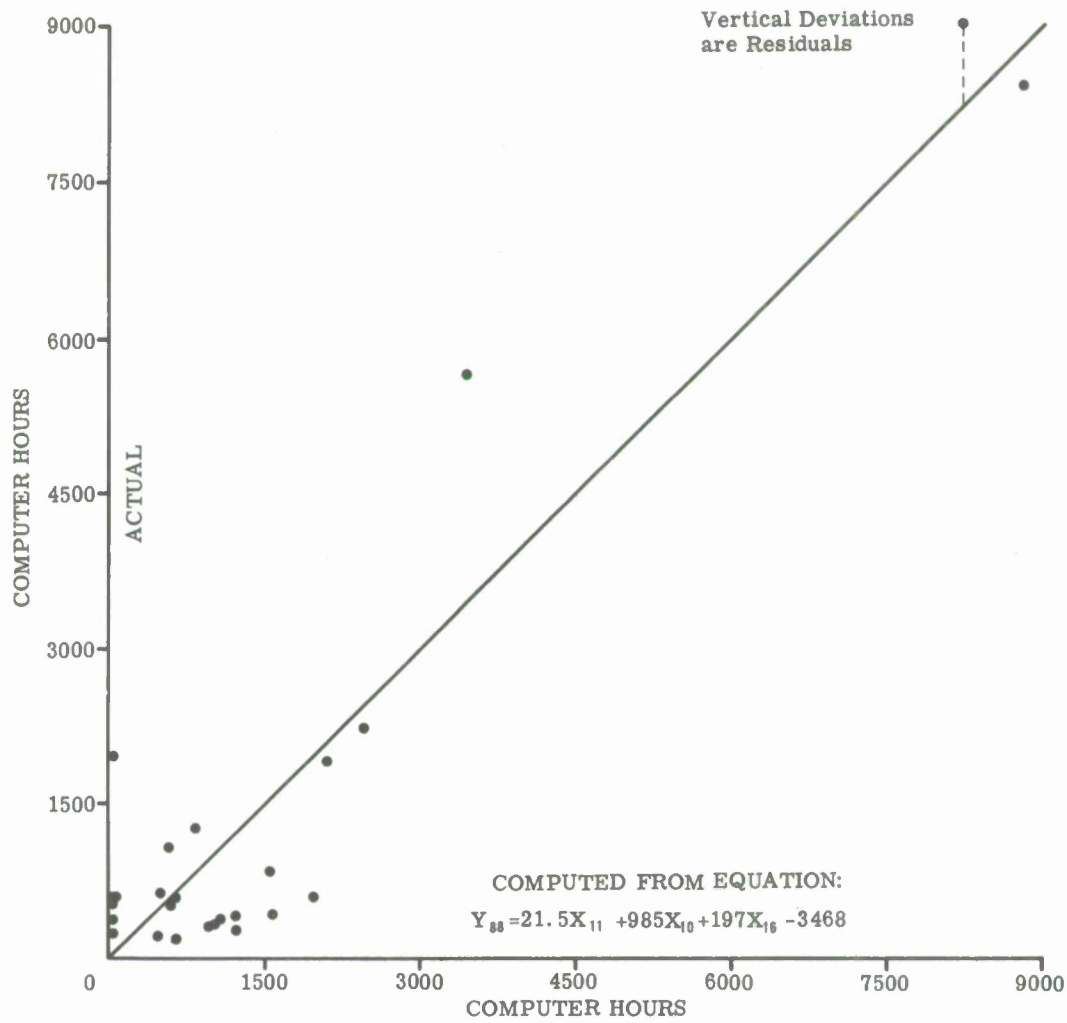


Figure 2. Actuals vs Computed for Cost Variable 88--Total Computer Hours (N = 26)

It is apparent that variables 10, 11, and 16 are components in both equations. In fact, an analysis of cost variable intercorrelations revealed that man months and computer hours had a correlation of .97; thus, it seems, one can be predicted from the other. Figure 3 provides a scatterplot and a simple regression equation showing the relationship between these variables:

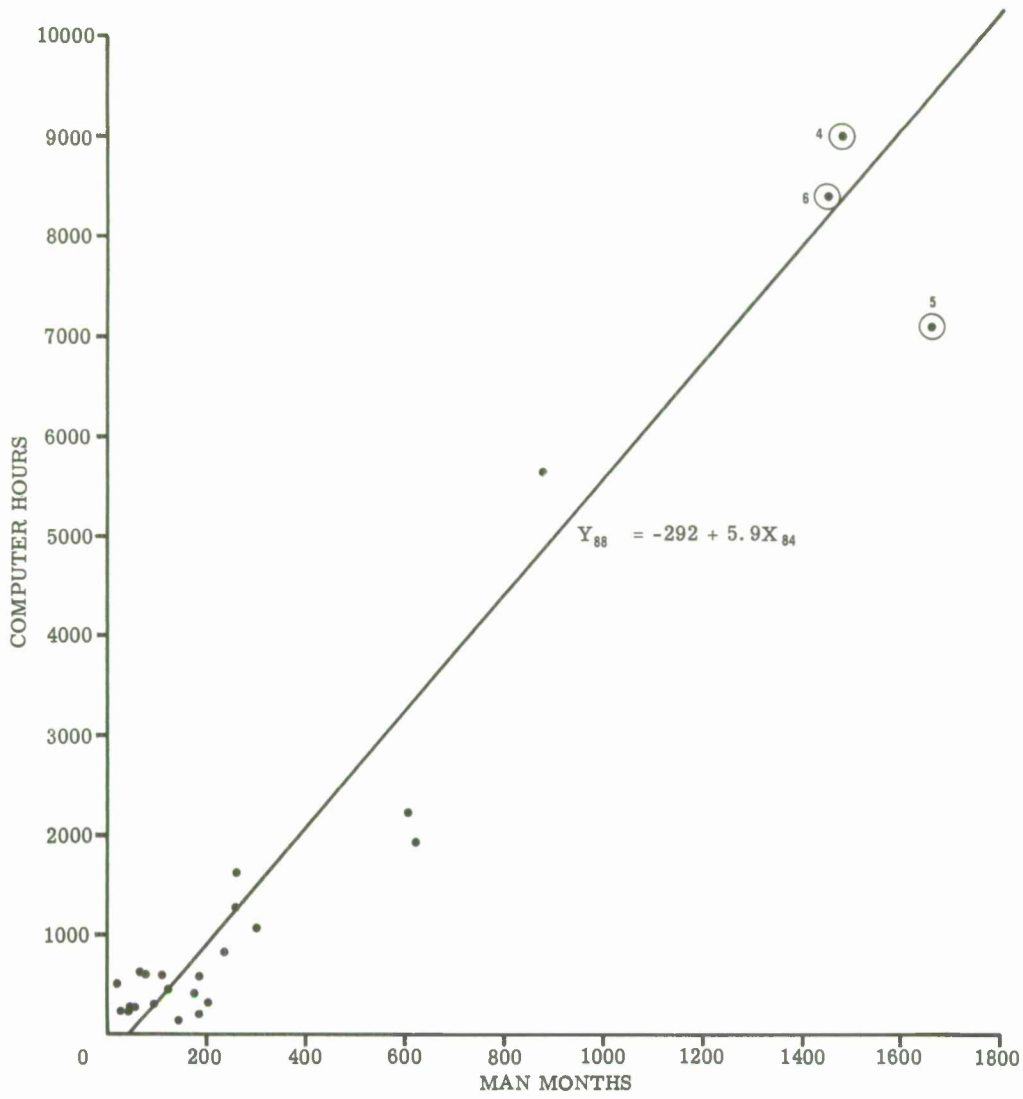


Figure 3. Total Computer Hours vs Man Months for Program Design, Code and Test (N = 26)

NOTE: Data point 5 is plotted here although it was not used in the derivation of the equation shown above.

The above relationship, if supported in continued analysis, implies that the problem of estimating computer time is reduced to the problem of estimating man months

Apparent in most of the early equations we derived was the dominance of predictor variable 11, estimated number of instructions, while other variables seemed to be playing relatively minor roles. We suspected that the remaining two large data points (4 and 6) in the sample were heavily influencing this condition due to their size and the accuracy with which they have been estimated. Therefore, we performed an additional analysis on cost variable 84 (man months), omitted data points 4, 5, and 6, and substituted variable 90 (delivered instructions) for variable 11 (estimated instructions). The results, shown below and detailed in Table V, do indeed demonstrate a decreased emphasis on number of instructions, and an increased significance of other variables:

$$Y_{84} = 2.8X_{90} + 1.3X_{83} + 33X_{39} - 17X_{59} + 10X_{46} + X_{12} - 188$$

Standard error of estimate = 70 M/M

95% confidence limit at the mean = $\pm$150 M/M

Variables

84	Man months for program design, code, and test
90	Delivered instructions (in thousands)
83	Trip mileage (thousands)
39	External document types
59	Type IV[1] programmer experience
46	Number of displays
12	Percent new instructions

A comparison was made of actual man months versus man months computed from the preceding equation. Figure 4 shows a marked decreased in the residuals, thus providing a visual illustration of the increased confidence that characterizes this equation.

[1]Type IV, the System Programmer, contributes to the formulation, planning, design, and development of large computer program systems. A more complete definition of programmer types is included on page 20 of the questionnaire.

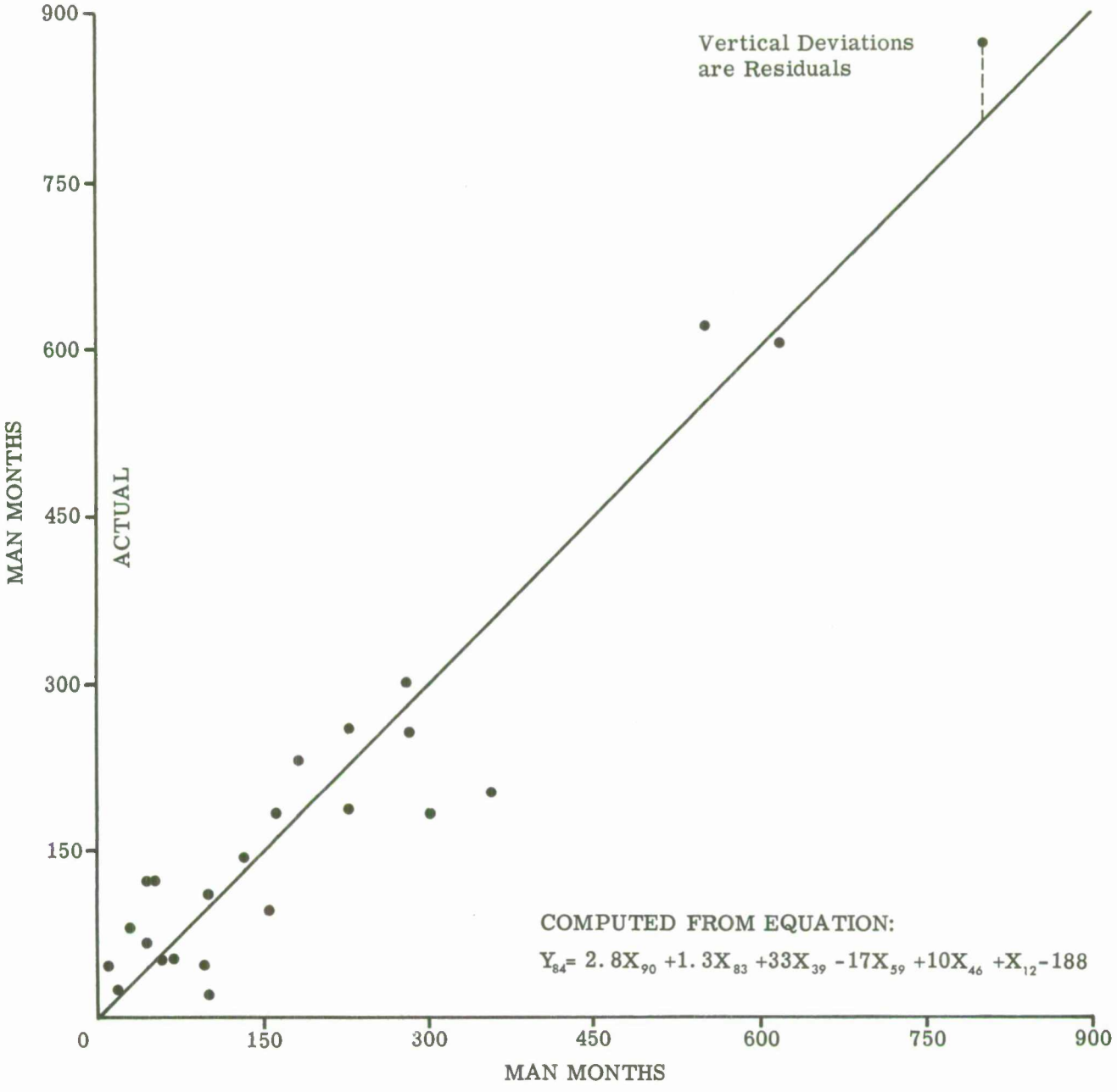


Figure 4. Actuals vs Computed for Cost Variable 84--Man Months for Program Design, Code and Test (N = 24)

NOTE: This was the final analysis, using delivered instructions in place of estimated instructions and excluding all extremely large programs.

TABLE V

SUMMARY OF CORRELATION AND REGRESSION ANALYSIS FOR COST VARIABLE 84

Man Months for Program Design, Code and Test

(Final analysis--using variable 90 and excluding all extremely large programs)

Variable Number	90	83	39	59	46	12	30	24
Short Description	Del'd Instr. (1000's)	Trip Mileage (1000's)	Ext. Docts. (Types)	T/4 Progr. Exper.	No. of Displays	% New Instr.	Prog. Chngs.	% Data Reduc. Instr.
Means	39.7	60.6	4.7	3.7	3.1	82.2	23.1	30.0
Standard Deviations	26.9	77.3	2.5	3.3	6.0	26.3	36.8	19.6
Validity Coefficients	.58	.68	.37	-.10	.68	.20	.67	-.22
Intercorrelations Variable Number								
90	1.00	.17	.17	.20	.54	-.14	.08	.13
83	.17	1.00	.14	-.02	.26	.15	.58	-.33
39	.17	.14	1.00	.52	.07	-.17	.42	-.42
59	.20	-.02	.52	1.00	-.14	-.39	-.09	-.29
46	.54	.26	.07	-.14	1.00	.05	.40	-.03
12	-.14	.15	-.17	-.39	.05	1.00	.01	.15
30	.08	.58	.42	-.09	.40	.01	1.00	-.32
24	.13	-.33	-.42	-.29	-.03	.15	-.32	1.00
Standardized Regression Coefficients (11 variables)*	.47	.34	.34	-.35	.26	.17	.12	-.19
Standardized Regression Coefficients (6 variables)	.35	.47	.38	-.27	.30	.12	not selected	not selected

Mean of Cost Variable	203	Number of Data Points	24
Multiple Correlation Coefficient	.96	Standard Deviation of Cost Variable	212
Standard Error of Prediction at the Mean	71	Standard Error of Estimate	70

95% Confidence Limits at the Mean** $\pm$150 Man Months

PREDICTION EQUATION: $Y_{84} = 2.8X_{90} + 1.3X_{83} + 33X_{39} - 17X_{59} + 10X_{46} + X_{12} - 188$

*There were 11 variables in the original selection run. Variables 26 (% Decision Instr.), 32 (Language Type) and 42 (Cptr. Oper. Doct'd) were also not selected due to extremely small standardized regression coefficients.

**These limits will expand as predictions deviate from the mean.

Instruction-Estimating Equation

Since the intermediate predictor variable, number of instructions, played such a significant role in this analysis, it is especially worthy of additional study. Even though the smaller sample (N = 24) analysis tended to reduce the contribution of this variable, a reliable technique is still needed to ascertain this quantity. This need is emphasized again in Figure 5, which depicts the relationship between man months and instructions, and Figure 6, which shows the relationship between computer hours and instructions.

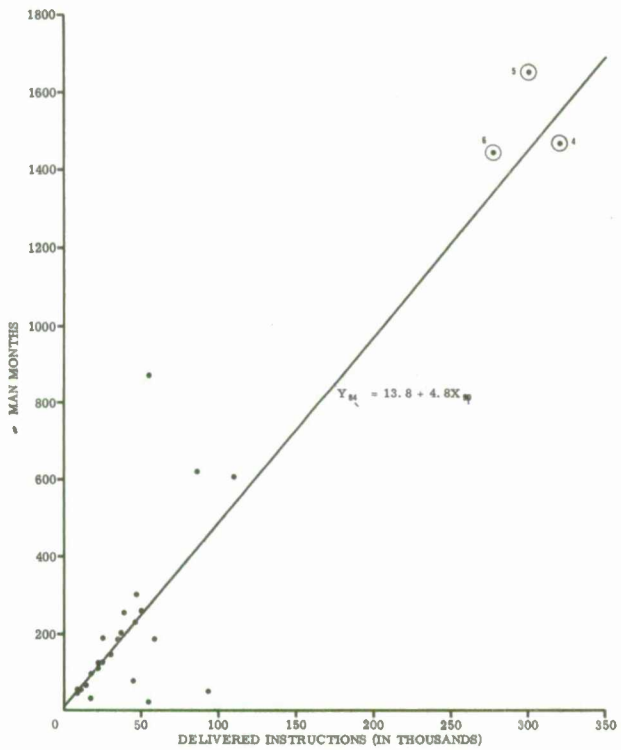


Figure 5. Man Months for Program Design, Code and Test vs Number of Delivered Program Instructions (N = 26)

NOTE: Data point 5 is plotted here although it was not used in the derivation of the equation shown above.

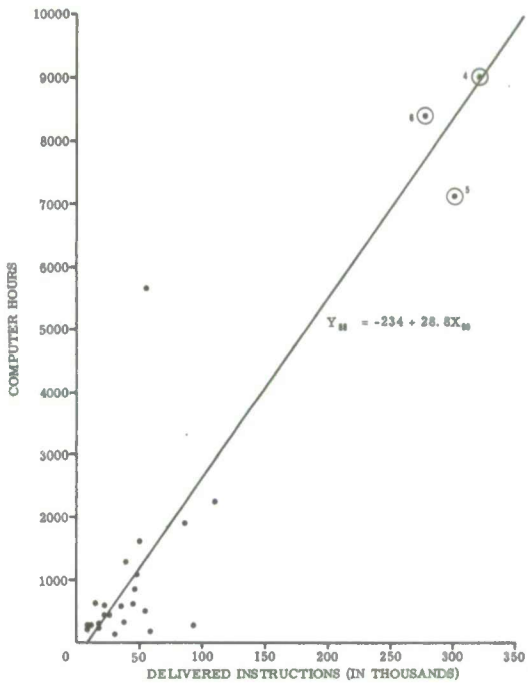


Figure 6. Total Computer Hours vs Number of Delivered Program Instructions (N = 26)

NOTE: Data point 5 is plotted here although it was not used in the derivation of the equation shown above.

Shown below are the results of a special analysis conducted to derive an equation for estimating the total number of delivered instructions without using the reported estimate of this number as a component in the equation.

$$X_{90} = 2.6X_{18} + 1.2X_{21} + 5.6X_{13} - 13.9$$

Standard error of estimate = 25.7 Inst. (Thousands)

95% confidence limit at the mean = $\pm$54.2 Inst. (Thousands)

Variables

90 Number of delivered instructions (in thousands)

18 Number of input message types

21 Number of subprograms

13 Number of words in tables and constants ($\log_{10}$)

Figure 7 shows a plot of actual versus computed number of instructions resulting from the application of the equation to 26 data points. Additional detail is provided in Table 8 of Appendix VII.

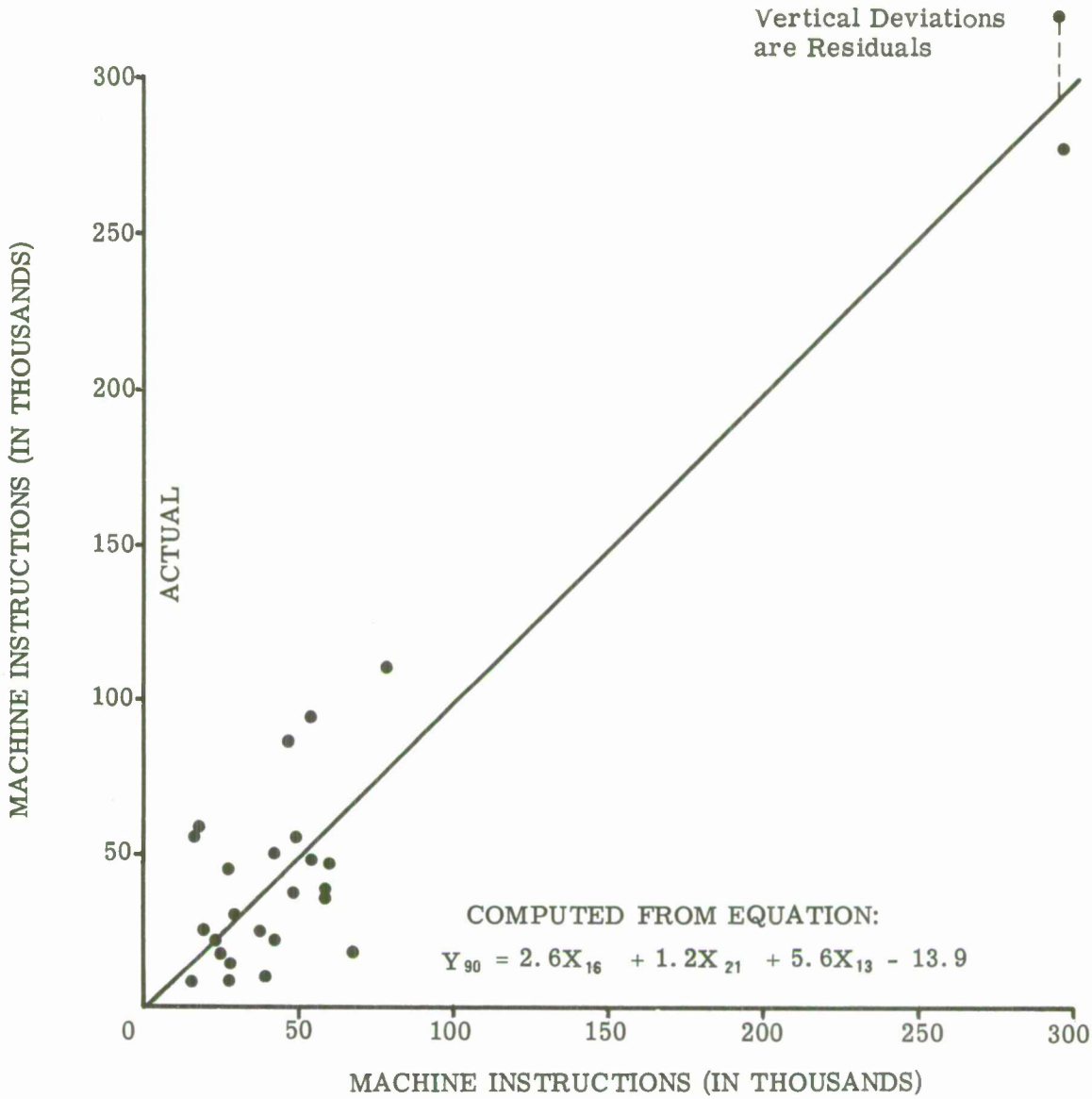


Figure 7. Actuals vs Computed for Cost Variable 90--Delivered Instructions (In Thousands) (Without Using Estimated Instructions, N = 26)

As might be expected, the preceding equation has rather broad confidence limits. We believe this condition stemmed, in part, from the original formulation of the present research. At that time, very little thought was given to variables that directly affect number of instructions, so that the predictions shown in that equation are not necessarily the most realistic indicators of this parameter. Other more fundamental factors must be formulated to describe more specifically the nature of the data processing task to be performed. At any rate, the dominance of number of instructions in this analysis provides a strong stimulus for a deeper investigation into the underlying factors associated with program size.

Summary and Conclusions

One relationship in which we can now begin to have increasing confidence is that between costs and the number of instructions in the completed program. In the sample of 26 data points (with data point 5 removed), cost, in terms of man months and computer hours, was primarily related to program size (instructions) and less influenced by other factors. Using reported estimates of instructions alone as a predictor of man months for program design, code and test, we obtained 95 percent confidence limits of 383 man months at the predicted mean of the cost variable. By adding rated program complexity, external document types, internal document types and number of data base words to the equation, the confidence limits were decreased, and the statistical confidence was increased by 23 percent. This suggested that the use of suitable predictor variables other than number of instructions would help to increase cost-estimating precision.

A substantial reduction in the 95 percent confidence limits for estimating man months was achieved by eliminating all the extremely large programs (data points 4, 5 and 6) from the regression and using variable 90 (delivered instructions) rather than variable 11 (reported estimated instructions) as a key predictor variable. This resulted in the selection of five companion predictor variables that provided an enhanced intuitive quality to the equation and increased the confidence in the final equation considerably. Specifically, the variables trip mileage, external document types, Type IV programmer experience, number of displays, and percent new instructions, when combined with delivered instructions, reduced the confidence limits to 150 man months, a reduction of 60 percent from those originally calculated. This is a strong indication that an appreciable increment in cost-estimating precision can be expected from the use of multiple predictor variables.

However, despite the dominance of number of instructions in our present research, it is only an intermediate cost-estimating parameter, not a measure of programming quality or program performance, and therefore, is not useful as a cost-effectiveness measure. To measure cost effectiveness, information concerning important but presently unmeasurable design characteristics such as a program's data-processing capability, complexity, reliability, usability and changeability will be needed.

It is not recommended that program development efforts be compared solely on the basis of cost per instruction. Perhaps an analogy to a more everyday example will make this important point clearer. A station wagon and a sports car that cost equal amounts may have an equal number of engine cylinders, but the value and performance of these two vehicles can be clearly distinguished in terms of fuel consumption, acceleration, design for family use or sports-car use, and so forth. Comparison of these cars on a cost-per-cylinder basis is virtually meaningless, which is our point concerning cost per instruction.

Since computer hours and man months were closely related in both the 26-data-point study ($r = .97$) and the 24-data-point study ($r = .91$), it is anticipated that similar findings will prevail concerning these major cost variables. Such findings, if substantiated in further studies, would provide a firm foundation for improving our cost-estimating techniques.

The results of the analysis of cost factors by statistical techniques illustrate clearly that meaningful relationships among both the factors and the costs can be derived. Such relationships can be determined with much more accuracy and validity by extending the analysis to larger samples of data and by probing more deeply into the fundamental nature of the data-processing task.

This study has been a first attempt to quantify the cost-contributing effects of some of the factors believed to affect programming costs. Work must be initiated in certain other areas if programming managers are to obtain a better understanding of the problems of costing, evaluating and comparing computer programs. The next section outlines some directions in which the present research may be extended.

V. RECOMMENDATIONS FOR FUTURE WORK

In addition to recommendations for a continuation of the cost analysis along the lines described in this report, we discuss here a number of problem areas appropriate for future research.

Systematic iteration of the activities of data collection and analysis is a necessary condition for achieving useful cost-estimating relationships. For example, many of the predictor variables rejected early in the study still hold great appeal and require further study to determine their utility in cost regression equations. Some of the rejected variables that have high logical appeal are listed in Table VI.

TABLE VI

SOME REJECTED VARIABLES THAT REQUIRE FURTHER STUDY

Variable Number	Correlation* with Man Months (84)	Short Variable Description	Comments
40	.88	Total number of computer hours per week	High correlation, but considered a feedback variable rather than a true predictor
19	.80	Number of output message types	Very highly correlated with 16, number of words in data base; 17, number of data base classes; and 18, number of inputs
30	.78	Number of program design changes	Difficult to estimate; correlated with 83, trip miles
17	.70	Number of data base classes ($\log_{10}$)	A possible alternate for 16, number of words in the data base ($\log_{10}$)
33	.56	Number of programming tools	Needs better description of tools; possibly a feedback variable
6	.34	Number of system design changes	Difficult to estimate
76	.30	Number of agencies required for concurrence	Seems to be tied to 77 and 78, experience and decision capability of agencies
32	-.30	Language type used	Possibly spurious algebraic sign
1	.17	Innovation in operational system	Needs better definition of innovation

*These coefficients, based on 26 data points, changed significantly when all the extremely large data points were removed. See the Validity Table for N = 24 in Appendix V.

We consider the techniques of regression analysis and factor analysis to be particularly robust and suitable tools with which to continue the research. As a result of the experience gained in this the first iteration, we feel that we have a sound basis for improving the initial design of the questionnaire and for collecting data to form a larger and more representative sample of program development. Specifically, in the immediate continuation of the cost analysis, we need a sample size of at least one hundred data points. This iteration has shown the feasibility of the basic approach; the next one, based on a sufficiently large sample, should result in estimating equations with higher reliability and validity.

Additional Techniques

In addition to the above recommendations for more satisfactory data collection and analysis, the continuation of our work might benefit from the application of the following techniques, which time did not permit us to use.

1. Using a Modified Step-Wise Regression Analysis to Select Predictor Variables. When the ratio of potential predictor variables to observed data points approaches or exceeds one (and the sample is relatively small) there is considerable risk that, as the population of variables is reduced to enable the computation of a meaningful regression function, some useful variables may be overlooked. One positive, although incomplete, method for reducing this risk is to select predictor variables by analyzing the correlation coefficients of all variables with the successive residuals resulting after the influence of the best single prior variable has been removed statistically. This approach is known as stepwise regression analysis and may be used successfully when a dependable and dominant predictor variable is available as a core around which to build the analysis (the variable called number of instructions appears to be this kind of a variable).

   Computer programs for conducting stepwise regression usually choose the highest partial validity coefficient at each successive step in the selection process. However, in a modified version of this approach, investigators can examine the results before each selection is made. In this way, the investigators may override the automatic selection when necessary and choose a selection sequence that best meets operational criteria. At the same time, they can also observe and tag promising predictor alternates for analysis by conventional regression procedures.

2. Questionnaire Item Analysis and Aggregation. One alternative available to minimize information loss, when there are many more predictor variables than data points in the sample, is the systematic aggregation of variables into homogeneous groups. This device is especially suitable when many of the variables are, in fact, dichotomous questionnaire items, i.e., items that can be answered YES or NO. If such items can be meaningfully scored

1 or 0, they can be grouped into submeasures that, in turn, could be handled as variables. Aggregation decreases the initial population of variables and thus allows a more favorable data point-to-variable ratio, while preserving more of the information in the questionnaire.

All dichotomous questionnaire items need not be aggregated--some may be ignored as a result of poor observed validity, i.e., low correlation with particular cost variables. Such items may be subject to exclusion from subsequent versions of the questionnaire on empirical grounds; however, contrary to the emphasis placed on eliminating redundant variables in regression analysis, items that are valid and highly intercorrelated may be kept to enhance the internal consistency and reliability of the questionnaire. In fact, the intercorrelations among items can be used to identify clusters that, in turn, help define the number and nature of the submeasures. Therefore, the use of item analysis and aggregation in follow-on research may lead to new and valid predictor variables.

3. Program Cluster Analysis by Using Inverted Factor Analytic Techniques. There are two major types of statistical factor analysis. One attempts to describe a complex of descriptive variables in terms of a reduced set of underlying factors. This is the conventional factor analysis and the one that was used to some extent in this research. There is another method of factor analysis, called inverted factor analysis or Q-Technique (13), which is concerned with the manner in which, not variables, but data points can be clustered and described more parsimoniously. The aim of such analysis would be to isolate and classify the basic types of computer program development efforts. Although inverted factor analysis was not employed in the current research, it appears to offer additional potential for determining whether programming systems can be grouped into homogeneous families and, therefore, it could become a valuable tool for investigating program system taxonomy.

Related Research Areas

Many other program development areas require research. In the following, we review briefly several of these. We feel that research here will be of inestimable value to programming managers and purchasers of programming products. In general, all of the suggestions are pointed toward providing a cost/value framework for managerial decision-making with respect to computer program development.

1. Development of Techniques for Estimating Program Size. Since, in this analysis, program size as measured by number of instructions had such a strong relationship to costs, a reliable technique to estimate size is sorely needed. One estimator, described in Figure 7, was developed by using regression analysis. However, this formula still has rather broad confidence limits. A related estimator was partially investigated and is described in an SDC document (Reference 14). In that document, a

well-defined relationship between program design requirements and number of program instructions was hypothesized. More specifically, the research hypothesized a relationship between the number of operational decisions contained in the program requirements and the number of decision class instructions, and then, in turn, a relationship between the number of decision class instructions and the total number of instructions. The former relationship has never been investigated; only the latter relationship was examined. However, the hypothesis of the relationship between decision class instructions and total instructions was tentatively supported in a frequency analysis of machine instructions.

2. Development of Techniques for Estimating the Cost of Programming Changes. Research to provide estimates for the cost of changes would be highly dependent on the results of work in improving cost-estimating techniques. Therefore, as a sequel to this study, an extension could be conducted to search for prediction methods that provide cost estimates in replanning, e.g., when changes in requirements are proposed. Since, in such cases, some program development work has already been done and the total job is more clearly defined, the predictions would have to be more accurate than those acceptable for an initial estimate. More details would be required in the statement of factors that influence the cost of changes. Additionally, better techniques would be needed to account for the requirement imposed by the need to modify work already completed.

3. Development of a Taxonomy of Computer-Based Information-Processing Systems. A basic need for managers, users, and researchers is a more systematic classification of both completed and projected work in information processing. With the rapid development of new tools, techniques, and applications in information processing, even the most advanced students in the field struggle to keep abreast of the technology. Part of this problem is the lack of a structure into which new developments can be placed to allow comparison with past efforts.

   To alleviate this problem, it would be necessary to develop a comprehensive taxonomy or a series of taxonomies. These classification schemes would provide generalized distributions (devoid of acronyms) along several dimensions, such as functions performed, design characteristics, development procedures, cost, elapsed time, and staffing. In addition to the intrinsic worth of such taxonomies for relating various information-processing developments, they could also provide a basis for collection of data concerning cost, performance and lead time for use in cost effectiveness studies. Additionally, they could possibly be used to develop a benchmark as an aid to improved qualitative comparison of the nonhardware portions of information-processing systems.

4. Development of Descriptors of Program Performance and Quality. In this task, researchers need to clarify, define, and determine measurements relating to the quality of computer programs and to program documentation. This area of work overlaps the cost work described previously as well as the effort toward an information-processing taxonomy.

   A deeper investigation of quality should consider:

   a. What programs are supposed to do and how they are intended to be used as reflected in requirements and design specifications.

   b. What programs actually do as determined by test, exercise and operational use.

   c. Ways in which desired quality, including performance characteristics, can be expressed unambiguously and preferably quantitatively, and how the products, both documents and programs, can be inspected during each programming activity to insure that quality standards are met.

   At present, the only measurable characteristics that are generally used to describe programs are computer operating time and program size or storage requirements. Although programs are classified by titles such as "storage and retrieval," or, at a lower level in the hierarchy, "input format conversion," there is no set of descriptors that permits easy comparison of programs for planning purposes and, more important, for cost estimation. In addition, there is the need to assign more meaning to expressions such as usability, modularity and maintainability as they apply to specific program design characteristics and as they apply to the way programs are used. The descriptors of performance and quality discussed here are intended to alleviate both the problem of unambiguous requirement specification and the quality control problem of testing programs so that errors can be efficiently detected and corrected.

5. Development of Cost Trade-offs and Cost/Value Relationships. The above studies of cost, quality and performance are all pointed toward cost-effectiveness analysis. In cost-effectiveness research, appropriately derived cost-estimating relationships and measures of quality and performance could be used to construct techniques that permit quantitative comparisons of proposed new products, tools and procedures. The research should seek the identification of preferred ways to develop and design nonhardware components based upon sound economic principles. For example, in computer program development, various trade-offs could be considered with respect to program design and performance, personnel mix, organization, scheduling, quality control practices, documentation design, and computer usage.

REFERENCES

(1) Anderson, R. L., and T. A. Bancroft. Statistical Theory in Research, New York, McGraw-Hill, 1952.

(2) Hald, A. Statistical Theory with Engineering Applications, New York, Wiley, 1952.

(3) Chew, V. (Editor). Experimental Designs in Industry, New York, Wiley, 1958.

(4) System Reliability Prediction by Function--Volume I, Development of Prediction Techniques, RADC-TDR-63-300, August 1963.

(5) Farrar, D. E., and R. E. Apple. "Some Factors that Affect the Overhaul Cost of Ships: An Exercise in Statistical Cost Analysis," Naval Research Logistics Quarterly, December 1963.

(6) Large, J. P. (Editor). Concepts and Procedures of Cost Analysis, RAND Memorandum RP-3589 PR, June 1963.

(7) Farr, L. A Description of the Computer Program Implementation Process, System Development Corporation TM-1021/002/00, 25 February 1963.

(8) Klein, L. R., and M. Nakamura. "Singularity in the Equation Systems of Econometrics: Some Aspects of Multicollinearity," International Economic Review, September 1962.

(9) McCornack, R., et al. Multiple Regression with Subsetting of Variables, System Development Corporation FN-6622/000/00, 11 June 1962.

(10) Harman, H. H. Modern Factor Analysis, Chicago, University of Chicago Press, 1960.

(11) Rogers, J. B. A-70A: 150 Variable Factor Analysis Program, System Development Corporation, TM(L)-863/001/00, 13 December 1962.*

(12) Rogers, J. B. Program A-26: Rotation of a Factor Matrix, System Development Corporation, TM(L)-863/003/00, 30 November 1962.*

(13) Cattell, R. B. Factor Analysis, New York, Harper and Bros., 1952.

(14) Bleier, R. E. Frequency Analysis of Machine Instructions in Computer Program Systems, System Development Corporation TM-1603, 19 November 1963.

*These documents were produced for limited circulation and are available only with the author's concurrence.

GUIDE TO APPENDICES

I QUESTIONNAIRE
Primary data gathering instrument. The cost factors of Volume I are rephrased in the form of questions in an attempt to quantify these variables.

II DEFINITION OF VARIABLES
The items in the questionnaire are then rephrased as the predictor and cost variables that are analyzed in this investigation.

III DATA MATRIX
The responses to the questionnaire are tabulated by variable and data point. Twenty-seven data points are described.

IV DATA ACCURACY
An assessment by the responders to the questionnaire of the accuracy accuracy with which 44 key questions were answered.

V VALIDITY TABLES
The correlations for all predictor variables with all cost variables are tabulated for both analyses of N = 26 and N = 24.

VI FACTOR LOADINGS
The results of the rotated factor loadings are tabulated for N = 26.

VII SUMMARY OF REGRESSION EQUATIONS
All the cost-estimating equations derived in this analysis are summarized and statistical details are tabulated such as the means, standard deviations, correlations, weighted and standardized regression coefficients, standard error of estimate, and confidence limits.

APPENDIX I--COST ANALYSIS QUESTIONNAIRE

INSTRUCTIONS FOR COMPLETING QUESTIONNAIRE

This questionnaire is a means for collecting data on past programming efforts. These data will help us to identify and verify key factors affecting the cost of computer programs. We are seeking to increase the reliability of techniques for estimating costs of program development.

The questionnaire is organized into seven parts. The first part, when completed, is an assignment sheet outlining the division of your program system or contract into program data points as defined below. A short description of each program corresponding to a data point is also requested. The six remaining parts are questions concerning some sixty-five factors that affect the cost of computer programs.

These factors have been organized into the following six parts.

- Operational Requirements and Design
- Program Design and Production
- Data Processing Equipment
- Programming Personnel
- Management Procedures
- Development Environment

Generally, speaking, the first two categories address the question, "What was the job to be done?" The next two ask, "What were the available resources?" and the last two examine, "What was the nature of the working environment?" Some of the factors may be specified or estimated readily by you, whereas many required that we develop arbitrary rules and definitions (since there are no standards), before these factors could be used. After each of the six categories of questions is a general question soliciting comments. Here we would be especially interested in any historical data that might have impact on the answers provided.

The information we are seeking is fairly detailed and most likely will not be readily available. Therefore, since some effort will be involved in compiling these data, we have attempted to make the questions as clear and definitive as possible. Even so, some of our definitions in the questionnaire are necessarily arbitrary and in some cases may be difficult to apply. We encourage answering all questions even if you have to redefine terms to suit the information available to you. When you find this to be necessary, please help us by giving a brief rationale for this change.

One problem in collecting data on computer programs is the definition of a program in terms of bounds on the program being examined. The definition leads us to the concept of a data point. We require the concept of a data point to standardize the definition of a program in order to better understand what it is we are trying to compare in our final analysis. The answers to this questionnaire will then allow the comparisons to be made on a more rigorous basis. One complete questionnaire is required for each program corresponding to a data point. We will need your help in identifying data points in accordance with the following definition.

A program data point is the smallest set of computer program instructions

(1) whose purpose is defined by someone other than the programmer,

(2) which is delivered to the user (customer) as a package, and

(3) which is loaded into the computer as a program unit or system to achieve the stated purpose or objective.

By this definition, a program data point can be an operational program, a utility program, or even an experimental program. These are clearly not limited to any specific function. Similarly, the user of the program (represented by the data point) may be the buyer, but he may also be another programmer, as in the case of a utility program. The responder must keep in mind at all times the portion of the program that he is calling the program data point when answering the questions. For example, a program data point as defined here could be a specific package in SATIN,* or a part of a model in SAGE* (e.g., Model 9, D.C.), or a phase in NORAD, an independent system such as ECAPS* in DODDAC or a subsystem in SACCS.*

Additionally, the definition of a program data point necessarily includes some clear statement of limits to the scope of activities considered as part of the programming process. Here, we are concerned with the activities of program design, code, test, and documentation.

A summary form is included to summarize the major costs of the program being examined in terms of man months, computer hours, and calendar time involved. Requested on this sheet, also, is a list of names of the persons to whom the various parts are delegated. A summary form is attached to each questionnaire.

Finally, we need your evaluation of the accuracy of the data presented. After each answer for which we require this evaluation, you will find an open parenthesis. By keeping the following table handy, you may conveniently fill in the parenthesis with one of the code numbers.

*SATIN--SAGE Air Traffic Integration
SAGE--Semi-Automatic Ground Environment
ECAPS--Emergency Capability System
SACCS--Strategic Air Command Control System

TABLE FOR ACCURACY VALUES

(to be inserted in open parenthesis as indicated in questionnaire)

From Records	From Memory	Judgment
1. Very accurate	4. Accurate recollection	7. Confident
2. Good estimate	5. Good guess	8. Good guess
3. Unreliable	6. Very hazy	9. Estimate

Your cooperation will be greatly appreciated. If there are any questions at all, please call L. Farr on Extension 439 in Santa Monica.

COST ANALYSIS QUESTIONNAIRE
Summary Form

Parts Delegated by _____

PROGRAM SYSTEM _____

PROGRAM DATA POINT _____

Names of responders to various parts:

PART A--OPERATIONAL REQUIREMENTS AND DESIGN _____

PART B--PROGRAM DESIGN AND PRODUCTION _____

PART C--DATA PROCESSING EQUIPMENT _____

PART D--PROGRAMMING PERSONNEL _____

PART E--MANAGEMENT PROCEDURES _____

PART F--DEVELOPMENT ENVIRONMENT _____

SUMMARY OF COSTS

1. Number of man months to design, code, test, and document the program including first-line supervision (Cat. I and Cat. II). _____ man months*
2. Number of man months of other labor such as secretarial and computer operator support (Cat. III). _____ man months
3. Average number of programmers employed on this program _____
4. Start date for program design. Month _____ Year _____

   Completion date for program delivery. Month _____ Year _____
5. Number of computer hours used by type of computer Type _____ Hours _____

   Type _____ Hours _____

   Type _____ Hours _____

*List all man month figures as effective man months.

6. Number of trips for briefings, problem solving, concurrence, etc. _____

Man months for travel _____

Average distance/trip _____

By program design, we mean the activity whose inputs are the operating system description and operational specifications; and whose outputs are program design specifications.

By program delivery, we mean the point at which the program is ready to be installed in the operational computer and begin its system test in the environment for which the program was designed.

Program Name _____

Responder _____

Date _____

A. OPERATIONAL REQUIREMENTS AND DESIGN

(Space is provided at the end of this section for comments or qualifying remarks on any question.)

1. Was there a requirement for innovation in the:	Check Appropriately		Reliability
. operational system,	Yes ____	No ____	( )
. hardware components.	Yes ____	No ____	

By innovation, we mean either a new application of a known technique and/or a new technique for a known application. By new, we mean new to the people involved.

Operational system refers to the objectives, mission, or functions to be performed.

Hardware components refers to data-processing equipment such as computer, I/O equipment, peripheral equipment, and automatic communication links.

2. Was there participation by the programming organization in the:			
. requirements analysis,	Yes ____	No ____	( )
. operational design.	Yes ____	No ____	( )

When the program is part of an information-processing system, the requirements analysis is conducted to specify in detail the performance requirements of this embedding system. These performance requirements are the input to the operational design activity, which translates the requirements into operational design specifications. These specifications indicate how the information-processing needs will be satisfied. If the program is not part of a larger information-processing system, then answer only the question concerning the operational design, which may include the requirements analysis.

3. How well were the operational requirements known:

In great detail _____ ( )

In broad outline _____

Only vaguely _____

4. How many system design changes were encountered? _____ ( )

During what part of the design phase did the peak load of changes occur?

Operational Design ____ ( )

Program Design ____ ( )

Here we mean system design changes in terms of functions, objectives, and components. Do not include here those changes required in the program system.

5. List the principal military commands and government agencies that interface with the program. _____ ( )

   Here, if there are any questions of security classification, list only the total number of agencies.

   _____

   _____

   _____

   _____

   _____

6. How many automatic data-processing centers are in the system? _____ ( )

   Here we mean computer-based centers for which computer programs were written.

7. How would you characterize the complexity of the information-processing system based on such factors as the number of interrelatedness of the functions to be performed, the components, and the users (not program system)?

   Rate on a scale of 1-5 from simple to highly complex. _____

8. Make any comments and qualifying remarks on any of your answers in this category (use reverse side or separate sheet if desired):

Program Name _____

Responder _____

Date _____

B. PROGRAM DESIGN AND PRODUCTION

(Space is provided at the end of this section for comments or qualifying remarks on any question.)

1. Indicate the number of computer program instructions and words (excluding the contents of the data base) as follows:

	Computer Name _____	Computer Name _____	Reliability
(a) Total number of instructions in original estimate.	_____	_____	( )
(b) Total number of instructions in delivered program.	_____	_____	( )
. New instructions written for this program.	_____	_____	
. Instructions reused from previous versions of this program.	_____	_____	
. Words in tables and constants.	_____	_____	
. Instructions resulting from the use of subroutines in a library.	_____	_____	
(c) Total number of instructions written but discarded and not delivered in the completed program.	_____	_____	

By instructions, we mean machine language instructions or orders. If the computer is a multi-address machine, count each instruction separately. If a procedure-oriented language is used, list the number of language statements and resulting machine instructions.

By subroutine, we mean some well-defined, logical or mathematical function.

By data base, we mean the subset of tables and words that describe the environment of the problem that the program is solving and/or the files to be processed.

2. Indicate the number of logical words (items) in the data base by:

   . total number of words, _____ ( )

   . number of classes of items. _____ ( )

Here, if the data base changes in size relatively often, list the range in size.

3. Indicate the volume of message throughout handled by the program by: ( )

By message type--we are not concerned here with the equipment involved, but are interested in the nature of the message, such as flight plan, position report, sensor data, weapon status, etc.

Here, if there are any questions of security classification, list the total number of message types without identification.

INPUT		OUTPUT	
Message Type	Average Rate per Unit Time	Message Type	Average Rate per Unit Time

(Use reverse of page if needed.)

4. Was there a requirement for innovation in the program design? Answer this on the basis of the design of the subprograms comprising the program:

. Total number of subprograms. _____ ( )

. List subprograms requiring innovation and number of instructions in each. _____ _____ instr. ( )

_____ _____ instr.

_____ _____ instr.

_____ _____ instr.

By innovation, we mean the same as in Question A1.
By subprogram, we mean the first level in the logical subdivision of data-processing functions in the program being considered.

Please keep in mind the program data point being considered. (Use other side of page if needed.)

5. (a) Was there a deliberate effort to include in the program design such characteristics as:

Check Appropriately

. maintainability, Yes _____ No _____ ( )

. usability, Yes _____ No _____

. (other) define. Yes _____ No _____

By maintainable, we mean the ease with which errors can be detected and corrected and new functions can be incorporated.

By usable, we mean the ease with which personnel other than the designer can use the program.

(b) If yes, by what technique was this characteristic specified (e.g., in design specification)?_____

_____

And how was its attainment tested?_____

_____

6. Characterize the complexity of the program design by indicating the percentage of the program (i.e., proportion of instructions) devoted to the following operational functions:

. clerical _____ ( )

. data reduction _____

. prediction _____

. decision making _____

Examples

Clerical - Read or write a table.
Data Reduction - Transform a coordinate or compute the average.
Prediction - Calculate a position from equations of motion (e.g., satellite tracking).
Decision Making - Make a logical choice given certain conditions (e.g., automatic weapons assignment). The number of conditional branches and requirements for self-modification are keys to this factor.)

7. Indicate the existence of constraints on program design, such as

. insufficient memory capacity	Yes _____	No _____	( )
. insufficient input/output capability	Yes _____	No _____	
. stringent timing requirements	Yes _____	No _____	
. other (specify)	Yes _____	No _____	

8(a) How many program design changes were encountered? _____ ( )

During what part of the production phase did the peak load of change occur?

Program design _____
Program code _____
Program test _____

By program design changes, we mean those program changes that resulted either from changes in operational functions or a change in design (thus permitting the same operational or data-processing function to be performed in a more efficient way).

(b) In implementing these program design changes, what was the cost in terms of:

. man months,	_____	( )
. computer hours,	_____	( )
. pages of documents.	_____	( )

(c) In evaluating and contemplating these program design changes and administering all necessary record keeping, what was the cost in terms of

man months _____ ( )

9. What language was used in coding the program? _____ ( )

10. Identify the type of programming tools available in developing this program:

Type	Available	Not Available	( )
Data Tools:			
Data Description Language	____	____	
Data Description Table Generation	____	____	
Data Description Table Design	____	____	
Format and List Data Description Tables	____	____	
Program Test Tools:			
Test Data Generation	____	____	
Test Data Insertion	____	____	
Test Data Recording	____	____	
Test Data Reduction	____	____	
Code Analysis	____	____	
Program Modification:			
Design Change Tools	____	____	
Error Correction Tools	____	____	
Parameter Change Tools	____	____	

	Available	Not Available
Control Tools:		
Support System Operating Control	____	____
Accounting or Bookkeeping	____	____
Labeling and Indexing I/O Media and Stored Data	____	____
Interruption, Intervention and Restart	____	____
Error Detection	____	____
Special Tools:		
Hardware Diagnostics	____	____
Program Parameter (data base) Generation and Manipulation (file maintenance, sequence parameter assembly and generation)	____	____
Output Editing and Formatting (report generators)	____	____
Loading Routines	____	____

11(a) Indicate the existence of test requirements and specifications for:

. parameter test, Yes ____ No ____ ( )

. assembly test Yes ____ No ____

By parameter test, we mean an evaluation of subprogram performance against the coding specifications. This activity is also called debugging or subprogram checkout.

By assembly test, we mean the verification that the component subprograms interact and communicate according to the program design specifications. This activity can be thought of as the program system test. Assembly test is conducted with simulated inputs in order to minimize the effects of people and equipment.

(b) If yes, how were these communicated?

orally _____

by document _____

(c) Did these specify the maximum range of inputs, outputs, and illegalities? Yes ____ No ____

(d) Did these specifications indicate the point at which to terminate testing (if other than the scheduled deadline)? Yes ____ No ____

If yes, how?

(e) List any performance measures that may have been specified (e.g., mean time between errors):

12. What was the requirement for documentation?

	Internal	External	
List types of documents	_____	_____	( )
	_____	_____	
	_____	_____	
	_____	_____	
	_____	_____	
	_____	_____	
	_____	_____	
	_____	_____	
	_____	_____	
	_____	_____	
	_____	_____	

(Use other side of page if necessary.)

Total number of pages (in one set) _____ _____

By types of documents, we mean the subjects and purposes of the documents, such as Operating System Description, Program Design Specifications, Status Reports, Error Correction Reports, Program Listings, etc.

By internal, we mean for the programming organization's use.

By external, we mean for delivery to the customer.

13. Make any comments and qualifying remarks on any of your answers in this category (use reverse side or separate sheet if desired):

Program Name _____

Responder _____

Date _____

C. DATA PROCESSING EQUIPMENT

(Space is provided at the end of this section for comments or qualifying remarks on any question.)

Reliability

1(a) Give name(s) of the computer(s) used in production of this program. _____ ( )

_____

_____

List primary computer first. Answer Questions 2(b)-6 on the basis of this computer.

(b) Give name(s) of the computer(s) used at the operational site if different from those listed above. _____

_____

(c) How many operational computer sites in system? _____

2(a) On the average, how many computer hours per week were scheduled for development of this program?

Shift	I (0800-1600)	II (1600-0000)	III (0000-0800)	( )
Computer				
_____				
_____				
_____				

(b) Was the available computer time considered adequate for:

. parameter test, Yes _____ No _____ ( )

. assembly test, Yes _____ No _____

(c) If no, how much more time was required? _____

3(a) How many hours per week were scheduled for preventive maintenance? _____ hrs/week ( )

(b) What per cent of scheduled available time was lost due to unscheduled maintenance for:

. the computer, _____%

. peripheral equipment. _____%

4(a) Were the operations of the computer and peripheral equipment adequately documented (e.g., error free, sufficiently clear and detailed, sufficient copies, etc.)? Yes _____ No _____ ( )

(b) If no, discuss briefly:

(c) Were the operations of the computer and peripheral equipment interrupted by equipment design changes? Yes _____ No _____ ( )

5. Describe the following computer characteristics:

- size of core storage, _____ words ( )
- size of secondary storage, _____ words
- word length, _____ bits
- add time (fixed point), _____ μs
- access time, _____ μs
- number of index registers, _____
- capability for handling bytes, Yes _____ No _____
- multiple or single address, Mult. _____ Single _____
- computer purchased, rented or provided. _____

6. Describe computer configuration:

- number of tape units, _____
- number of buffer drums, _____
- number of disc units. _____

7. How many automatic data-processing components were developed concurrently with this program? _____ ( )

   By ADP components, we mean those pieces of equipment that are somehow recognized, addressed, or controlled by the computer program (e.g., displays, message composers, converters, etc.).

*8(a) List the number and types of displays driven by this program. ( )

(b) List the number and types of input/output equipment (e.g., tapes, typewriters, etc.).

(c) List the number and types of input equipment.

(d) List the number and types of output equipment.

*Where lists are requested, and detailed information is unavailable, please estimate the total number of items that would appear in the list.

*9(a) List the pieces of EAM equipment that were available for this program development effort.

(b) Were these considered adequate? Yes _____ No _____

10. Make any comments or qualifying remarks on any of your answers in this category (use reverse side or separate sheet if desired).

*Where lists are requested, and detailed information is unavailable, please estimate the total number of items that would appear in the list.

Program Name _____

Responder _____

Date _____

D. PROGRAMMING PERSONNEL

(Space is provided at the end of this section for comments or qualifying remarks on any question.)

1. Fill in the following table with average values:

   . number of programmers by type
   . years of experience with language
   . years of experience with computer
   . years of experience with application.

I	II	III	IV

Reliability
( )

Type	Position	Description
I	Coder	Writes machine language instructions from flow charts. Helps prepare flow charts and test programs.
II	Programmer	Develops programs to solve well-defined problems. Prepares flow charts, writes instructions, tests programs, modifies established computer programs.
III	Senior Programmer	Conceives, develops and improves large, complex computer programs, e.g., automatic programming routines. Improves efficiency of existing programs.
IV	System Programmer	Formulates and plans new program system applications. Keeps abreast of related economic disciplines and new information processing technology. Is highly creative in designing and developing major computer program systems.

D. PROGRAMMING PERSONNEL

2. Indicate in the table below: Reliability

(a) How many programmers and what type participated in the operational design?

(b) Of the above, how many participated in the program design?

(c) What was the total number of programmers participating in the program design?

(d) How many programmers worked on the program for the entire duration of the project?

(e) How many programmers (on the average) terminated per month?

(f) During the first few months, how many programmers (on the average) were hired per month?

( )

Type*

	I	II	III	IV
2(a)				
2(b)				
2(c)				
2(d)				
2(e)				
2(f)				

(g) How many other personnel were required in the production of this program? _____

3. Make any comments and qualifying remarks on any of your answers in this category.

*Type defined on page 20.

Program Name _____

Responder _____

Date _____

E. MANAGEMENT PROCEDURES

(Space is provided at the end of this section for comments or qualifying remarks on any question.)

Check one: Reliability

1. Was the programming organization function- or project-oriented? Project __ Function __ ( )

   By function-oriented, we mean that programmers performed only specialized functions, such as design, code, or test.

   By project-oriented, we mean the same programmers carried the job through from design through test.

2. Was there a documented plan or procedure for the following:

	No	Yes	If yes, document number
(a) Evaluation and implementation of system design changes			
(b) Evaluation and implementation of program design changes			
(c) Dissemination of error-detection and error-correction information			
(d) Use of the computer facility in terms of number of runs per day per programmer. If yes, how many? _____			
(e) Contigency plan in the event the computer was overloaded or otherwise unavailable			
(f) Communicating with other agencies (e.g., those listed in A5 and F1)			
(g) Concurrence on design specifications			
(h) Cost control			
(i) Management control in the form of PERT or Gantt charts			
(j) Document control (e.g., design file)			
(k) Standards for coding, flow charts, etc.			

E. MANAGEMENT PROCEDURES

3. Make any comments and qualifying remarks on any of your answers in this category.

Program Name _____

Responder _____

Date _____

F. DEVELOPMENT ENVIRONMENT

(Space is provided at the end of this section for comments or qualifying remarks on any question.)

Reliability
( )

1. List the agencies whose concurrence was required on design specifications in the table below.

   (a) Were the personnel representing this agency experienced in the development of information processing systems?

   (b) Of those with experience, were any decision makers?

Agency Name	Indicate Yes or No	
	Question (a)	Question (b)

F. DEVELOPMENT ENVIRONMENT

Reliability

2(a) How many man-months were devoted to concurrence? _____ ( )

(b) Did the schedule slip because of lack of timely concurrence? Yes_____ No _____

(c) If yes, how much and describe causes

3. Was the computer operated by an agency other than the program developer? Yes _____ No _____ ( )

4(a) Was the program developed at a site other than the operational location Yes _____ No _____ ( )

(b) Did the program development take place at more than one location during the effort? Yes _____ No _____

5. Make any comments or qualifying remarks on any of your answers in this category.

APPENDIX II--DEFINITION AND CODING OF VARIABLES

INTRODUCTION

This Appendix defines the independent (predictor) variables and the dependent (cost) variables for which data were collected by means of the questionnaire (Appendix I). The first column indicates the source question in the questionnaire that requests some measure on the variable. The second column is a brief description of the variable and third column identifies the variable by a number for data processing purposes. The last column shows how the response to the question was coded in the event a nonquantitative answer was required.

APPENDIX II. DEFINITION AND CODING OF VARIABLES

Question Number	Independent Variable	Variable Number	Coding
A			
1	Innovation in Operational System	1	Yes = 1 No = 0
	hardware components	2	Yes = 1 No = 0
2	Participation in Requirements Analysis	3	Yes = 1 No = 0
	Operational Design	4	Yes = 1 No = 0
3	How well Operations Requirements known	5	Great detail = 3 Broad outline = 2 Vaguely = 1
4	Number of System Design changes	6	_____
	Peak load of changes occurred	7	Operations Design = 1 Program Design = 2
5	Number of commands	8	_____
6	Number of ADP centers	9	_____
7	Complexity rating	10	_____
B			
1(a)	Number of instructions in original estimate	11	Divided by 1000
(b)	Ratio: new instructions/delivered instructions	12	_____
	Ratio: reused instructions/delivered instructions	-	_____
	Number words in tables and constants	13	$\log_{10}$
	Ratio: subroutine instructions/delivered instructions	14	_____
(c)	Ratio: discarded instructions/delivered instructions	15	_____
2	Number of words in data base	16	$\log_{10}$
	Number of classes of items	17	$\log_{10}$
3	Number of input messages	18	_____
	Number of output messages	19	_____
4	Ratio: innovation instructions/delivered instructions	20	_____
	Number of subprograms	21	_____
5(a)	Effort to include characteristic of maintainability	22	Yes = 1 No = 0
	usability	-	Yes = 1 No = 0
6	Percentage of instructions clerical	23	_____
	data reduction	24	_____
	prediction	25	_____
	decision making	26	_____

Question Number	Independent Variable	Variable Number	Coding
7	Program design constraints: insufficient memory	27	Yes = 1 No = 0
	insufficient I/O	28	Yes = 1 No = 0
	timing	29	Yes = 1 No = 0
8(a)	Number of program design changes	30	_____
	Peak load of changes occurred	31	Program Design = 1 Program Code = 2 Program test = 3
9	Language type used	32	Compiler = 1 Assembly = 2
10	Number of programming tools	33	_____
11(a)	Existence of test requirements and specifications for parameter test	34	Yes = 1 No = 0
	assembly test	35	Yes = 1 No = 0
(b)	How communicated	-	Oral = 1 Document = 2
(c)	Specify inputs, outputs, etc.	36	Yes = 1 No = 0
(d)	Specify termination point	37	Yes = 1 No = 0
12	Number of internal documents	38	_____
	Number of external documents	39	_____
C			
1(c)	Number of computer sites in system	-	_____
2(a)	Total number of hours per week	40	_____
(b)	Was computer time adequate for parameter test	41	Yes = 1 No = 0
	assembly test	-	Yes = 1 No = 0
3(a)	Number of hours/week for preventive maintenance	-	_____
(b)	Per cent of time lost due to unscheduled maintenance for the computer	-	_____
	peripheral equipment	-	_____
4(a)	Computer operations adequately documented	42	Yes = 1 No = 0
(c)	Computer operations interrupted by equipment design changes	43	Yes = 1 No = 0
5	Number of words in core storage	44	divided by 1000
7	Number of ADP components developed concurrently with program	45	_____

Question Number	Independent Variable	Variable Number	Coding
8(a)	Number of displays	46	_____
(b)	Number of pieces of input/output equipment	47	_____
(c)	Number of pieces of input equipment	48	_____
(d)	Number of pieces of output equipment	49	_____
9(a)	Number of pieces of EAM equipment	50	_____
(b)	Were these considered adequate	51	Yes = 1 No = 0
D			
1	Number of programmers by Type I*	52	_____
	Number of programmers by Type II	53	_____
	Number of programmers by Type III	54	_____
	Number of programmers by Type IV	55	_____
	Yrs of exp with computer, language & application Type I	56	_____
	Yrs of exp with computer, language & application Type II	57	_____
	Yrs of exp with computer, language & application Type III	58	_____
	Yrs of exp with computer, language & application Type IV	59	_____
2(a)	Ratio: Operations Design programmers/total programmers	60	_____
(b)	Ratio: Program Design programmers(who participated)/ Operations Design programmers	61	_____
(c)	Ratio: Program Design programmers (total)/total programmers	62	
(d)	Ratio: Programmers (continual)/total programmers	63	_____
(e)	Number of terminations per month	64	_____
(f)	Number of hires per month	65	_____
E			
1	Programming organization	-	Function = 1 Project = 2
2(a)	Document for: system design changes	66	Yes = 1 No = 0
(b)	program design change	67	Yes = 1 No = 0
(c)	error detection and correction	-	Yes = 1 No = 0
(d)	use of computer facility	68	Yes = 1 No = 0
(e)	unavailable computer	69	Yes = 1 No = 0
(f)	communication with other agencies	70	Yes = 1 No = 0
(g)	concurrence procedures	71	Yes = 1 No = 0
(h)	cost control	72	Yes = 1 No = 0
(i)	management control	73	Yes = 1 No = 0
(j)	document control	74	Yes = 1 No = 0
(k)	standards	75	Yes = 1 No = 0

* Programmer types are defined on pg 20 of the questionnaire, Appendix I

Question Number	Independent Variable	Variable Number	Coding
F			
1	Number of agencies required for concurrence	76	_____
(a)	Ratio of experienced agencies/total agencies	77	_____
(b)	Ratio of decision-making agencies/total agencies	78	_____
2(b)	Schedule slippage due to lack of concurrence	79	Yes = 1 No = 0
3	Computer operated by another agency	80	Yes = 1 No = 0
4(a)	Program developed at site different than operational location	81	Yes = 1 No = 0
(b)	Development effort at more than one location	82	Yes = 1 No = 0
Summary 6	Number of trips x average miles/trip	83	Divided by 1000

Question Number	Dependent Variable	Variable Number	Coding
Summary			
1	Number of man months to design, code and test	84	
2	Number of man months for other labor	85	_____
3	Number of programmers	86	_____
4	Number of months of elapsed time	87	_____
5	Number of computer hours	88	_____
6	Number of man months for travel	89	_____
			_____
B1(a)	Number of instructions in delivered program	90	
B8(b)	Number of man months for program design change	91	
B8(b)	Number of computer hours for program design change	92	_____
B8(b)	Number of pages of documents for program design change	93	_____
B8(c)	Number of man months for evaluating program design change	94	_____
B12	Number of pages of internal documents	95	Divided by 100
B12	Number of pages of external documents	96	Divided by 100
D2(g)	Number of other personnel	97	_____
F2(a)	Number of man months for concurrence	98	_____
	Sum of Variables 84, 85, 89, 98	99	
	Sum of Variables 91, 94	100	

APPENDIX III--DATA MATRIX

INTRODUCTION

Data collection was conducted by means of the questionnaire in Appendix I. Each questionnaire, of which twenty-seven were completed, serves as a "data point."* The responses to the questionnaire are reported in this Appendix in a matrix form. The data may differ from those in the completed questionnaire for the following reasons:

(1) Data rounding and scaling.

(2) Transformation to percentages or ratios.

(3) Transformation to logarithms.

(4) Modification as a result of a conversation with the responder.

(5) Omissions, where guesses were not made, were estimated by the researchers.

This last point, (5), deserves additional comment. The computer program which is used for the regression analysis is not designed to handle missing data. Therefore, we used our judgment and experience to estimate the missing values. These estimated values are identified by a parenthesis in the data matrix.

The row headings in Appendix III identify the "data points" or programs being studied and the column headings are highly abbreviated descriptions of the variables. Appendix II, a more complete definition of the variables and their associated coding, includes (1) the source question in the questionnaire (Appendix I), (2) the variable number, and (3) the coding for the variables for use in the statistical analysis performed by the computer programs. Variables eliminated before the first computer run have no variable number assigned.

---

*A "data point" is the smallest set of instructions:
(1) whose purpose is defined by someone other than the programmer,
(2) which is delivered to the user as a package, and
(3) which is loaded into the computer as a program unit or system to achieve the stated purpose or objective.

Variable Number →		New System 1	New Hardware 2	Partic. in Req. Anal. 3	Partic. in Oper. Design 4	How well Ops Req'ts Known 5	No. System Changes 6	Time Peak Changes 7	No. of Commands 8	No. of ADP Centers 9	Complexity 10
Data Point	1	1	0	1	1	2	4	2	2	1	3
	2	1	0	0	1	3	15	2	4	1	4
	3	1	0	1	1	3	0	0	4	1	3
	4	1	0	0	1	2	10	2	4	2	5
	5	1	1	1	1	1	0	0	2	2	3.5
	6	1	0	1	1	2	50	2	4	2	5
	7	1	1	0	1	3	31	2	4	1	3
	8	1	0	1	1	3	75	2	4	1	3
	9	1	0	1	1	3	50	2	4	1	3
	10	1	1	0	1	1	5	2	1	2	4
	11	0	0	0	0	3	0	0	2	2	3
	12	1	1	0	1	2	1	1	2	2	2
	13	0	0	1	1	3	10	2	2	2	3
	14	0	1	0	1	3	30	2	2	2	3
	15	1	0	1	1	1	2	2	1	1	3
	16	0	0	0	1	3	1	2	1	1	3
	17	1	0	0	0	3	1	2	1	1	3
	18	1	1	1	1	2	5	1	0	14	3
	19	1	1	1	1	2	12	1	1	1	2
	20	0	1	0	1	2	10	1	4	34	5
	21	0	1	0	0	2	0	0	1	34	2
	22	1	1	1	1	3	4	2	5	34	4
	23	0	1	1	1	3	5	1	5	34	4
	24	0	1	1	0	3	5	2	5	34	2.5
	25	1	1	0	0	3	4	2	5	34	2.5
	26	0	0	1	1	1	(0)	(0)	(0)	1	3
	27	0	0	1	1	1	5	2	6	1	3

Variable Number →		Est. Instr. (thousands) 11	% New Instr. 12	% Reused Instr. X	Log, No. Words in Tables, Constants 13	% Subroutine Instr. 14	% Discarded Instr. 15	Log, No. Words Data Base 16	Log, No. Classes in Data Base 17
Data Point	1	41	100	0	4.83	9	0	0.00	0.00
	2	66	100	0	4.50	10	0	0.00	0.00
	3	71	12	88	4.60	11	0	0.00	0.00
	4	238	97	3	7.81	7	53	8.19	3.76
	5	150	92	8	0.00	0	33	0.00	0.00
	6	270	96	4	7.08	3	0	7.68	3.91
	7	20	60	40	4.28	0	58	3.64	0.85
	8	28	80	20	4.32	0	0	3.66	0.84
	9	16	40	60	4.34	0	0	3.70	0.85
	10	14	100	0	3.56	0	30	0.00	0.00
	11	43	100	0	1.00	0	0	0.00	0.00
	12	17	100	0	2.60	0	0	(4.60)	2.12
	13	15	100	0	3.48	0	6	6.04	2.01
	14	40	47	53	4.40	19	0	5.98	2.02
	15	60	100	0	4.30	0	300	2.18	0.30
	16	10	48	52	0.00	0	0	4.00	1.30
	17	15	89	0	0.00	11	0	0.00	0.00
	18	14	62	38	2.90	35	30	0.00	0.00
	19	151	95	5	4.17	73	59	3.48	0.61
	20	60	100	0	4.10	0	73	3.54	0.60
	21	45	100	0	3.70	0	20	0.00	0.00
	22	16	59	41	4.93	0	0	1.40	0.78
	23	30	92	8	4.08	0	0	2.40	0.78
	24	15	100	0	3.60	0	0	2.18	0.70
	25	12	100	0	4.00	0	0	2.18	0.78
	26	4	100	0	3.30	0	10	0.00	0.00
	27	7	100	0	3.30	0	10	5.60	1.48

Variable Number →		No. Input Messages 18	No. Output Messages 19	% Instr. Requiring Innovation 20	No. Sub-programs 21	Maintain-ability 22	Usability X	% Clerical Instr. 23	% Data Reduction Instr. 24	% Prediction Instr. 25
Data Point	1	0	0	21	3	1	1	20	58	2
	2	2	3	0	27	1	1	19	11	45
	3	2	3	84	30	1	1	20	25	45
	4	73	80	59	62	1	1	27	23	15
	5	0	0	27	2	1	1	5	20	5
	6	49	85	100	120	1	1	10	20	20
	7	3	0	30	35	1	1	28	17	25
	8	3	6	41	35	1	1	40	15	15
	9	3	6	87	31	1	1	35	15	15
	10	21	14	74	5	0	0	10	0	0
	11	0	0	0	22	0	0	30	40	10
	12	5	18	100	13	1	1	50	40	0
	13	9	22	0	16	0	0	10	60	10
	14	10	50	0	20	0	0	30	60	0
	15	1	1	100	5	1	1	75	0	0
	16	4	8	0	21	0	0	70	15	0
	17	8	6	0	12	1	1	50	50	0
	18	0	0	35	19	1	1	60	20	0
	19	16	11	24	23	1	1	5	50	5
	20	6	9	15	21	1	1	15	5	25
	21	0	0	100	30	1	1	40	20	0
	22	4	0	30	16	1	1	10	20	30
	23	3	0	10	18	1	1	10	60	10
	24	4	0	51	9	1	1	10	20	30
	25	6	9	35	13	1	1	10	60	10
	26	0	1	0	14	0	0	80	20	0
	27	2	3	0	16	1	0	50	20	0

Variable Number →		% Decision Making Instr. 26	Insufficient Memory 27	Insufficient I/O 28	Timing Constraint 29	No. of Program Changes 30	Time Peak Program Changes 31	Language Type 32	No. of Program Tools 33
Data Point	1	20	0	0	0	3	2	1.0	6
	2	25	1	1	1	(20)	3	1.0	16
	3	10	1	0	1	(0)	3	1.0	17
	4	35	1	1	0	136	3	1.0	19
	5	70	0	1	1	50	1	1.0	8
	6	50	1	1	1	100	2	1.0	17
	7	30	0	1	1	30	3	2.0	17
	8	30	0	1	1	70	3	2.0	17
	9	35	0	1	1	45	3	2.0	17
	10	90	0	1	1	25	3	2.0	11
	11	20	0	0	0	0	0	2.0	5
	12	10	0	1	0	50	2	1.0	10
	13	20	1	1	1	15	1	2.0	15
	14	10	1	1	0	50	2	1.0	12
	15	25	0	0	0	2	3	1.0	13
	16	15	1	1	1	20	3	1.0	(10)
	17	0	0	0	0	(0)	(0)	1.0	9
	18	20	0	1	0	(5)	1	1.5	11
	19	40	1	0	1	7	2	1.0	16
	20	55	1	1	1	170	2	2.0	14
	21	40	0	0	0	(0)	(0)	2.0	2
	22	40	1	0	1	5	1	2.0	9
	23	20	1	0	1	5	1	2.0	6
	24	40	1	0	0	4	1	2.0	6
	25	20	1	0	0	5	1	2.0	6
	26	0	0	0	0	4	2	2.0	7
	27	30	1	0	0	20	1	2.0	7

Variable Number →		Parameter Test Req'ts 34	Assembly Test Req'ts 35	How Communicated X	Req'ts Specify Inputs Outputs 36	Req'ts Specify When Stop Testing 37	No. Internal Document Types 38	No. External Document Types 39
Data Point	1	0	0	0	0	0	1	1
	2	1	1	2	1	1	1	9
	3	0	1	2	1	1	1	5
	4	1	1	2	1	1	10	13
	5	0	0	2	0	0	5	5
	6	1	1	2	1	1	8	19
	7	0	1	0	0	0	1	9
	8	0	0	0	0	0	1	9
	9	0	1	2	0	0	1	9
	10	0	1	2	1	0	0	6
	11	1	1	2	0	1	4	5
	12	0	1	2	1	1	5	7
	13	0	1	2	1	1	4	2
	14	0	1	2	1	0	3	5
	15	1	1	2	1	1	1	5
	16	0	1	2	0	0	(2)	2
	17	1	1	2	0	0	4	3
	18	0	1	2	0	1	5	3
	19	1	1	2	0	0	11	3
	20	0	0	0	0	0	5	6
	21	1	0	1	0	0	6	3
	22	1	1	2	1	0	1	4
	23	1	1	2	1	0	1	3
	24	1	1	1	1	0	1	5
	25	1	1	1	1	0	0	5
	26	1	1	2	1	0	11	2
	27	1	1	2	1	0	11	2

Variable Number →		No. Computer Sites X	Computer Hrs. per week 40	Computer time adequate--parameter test 41	Assembly Test X	Hrs/Wk for Prev. Maint. X	Computer Oper. Adeq'ly Documented 42	Computer Design Interrupt 43	Core Size (thousands) 44
Data Point	1	1	7	1	1	5	1	0	32
	2	1	50	1	0	15	1	0	32
	3	1	12	1	1	15	1	0	32
	4	2	150	0	0	-	0	1	65
	5	2	25	1	1	28	1	1	65
	6	2	80	0	0	-	0	1	65
	7	1	25	0	0	24	1	1	69
	8	1	35	1	1	24	1	1	69
	9	1	45	1	1	24	1	0	69
	10	2	9	0	1	10	0	0	32
	11	2	4	1	1	8	1	1	32
	12	2	4	1	1	8	1	1	32
	13	2	30	0	1	-	1	0	32
	14	2	13	0	0	7	0	1	32
	15	2	3	0	1	-	0	1	32
	16	-	18	0	0	-	(0)	1	16
	17	2	20	1	1	-	1	1	32
	18	14	17	1	1	-	1	0	69
	19	1	49	1	1	-	1	0	32
	20	34	31	1	1	10	0	1	24
	21	34	17	1	1	20	1	1	24
	22	34	5	1	1	10	1	1	24
	23	34	5	1	1	10	1	1	24
	24	34	13	1	1	10	0	0	24
	25	34	9	1	1	10	0	1	24
	26	1	10	1	0	12	1	1	8
	27	1	20	1	0	12	1	1	8

Variable Number →		No. EDP Components Developed 45	No. of Displays 46	Input Output Equip. 47	Input Only Equip. 48	Output Only Equip. 49	Pieces EAM Equip. 50	EAM Adequate 51	% Comp. Time Lost Unsch. Maint. X	% Time Lost Peripheral Equip. X
Data Point	1	0	0	14	1	2	5	1	-	-
	2	2	1	21	0	2	5	1	5	10
	3	2	1	21	0	2	5	1	-	-
	4	0	0	11	(0)	(0)	(5)	1	-	-
	5	0	0	2	2	2	(5)	0	5	35
	6	0	0	(2)	(2)	(2)	(5)	1	-	-
	7	4	6	41	8	13	5	1	5	5
	8	0	6	41	8	13	5	1	5	5
	9	0	6	41	8	13	5	1	5	5
	10	1	0	15	2	3	6	0	5	5
	11	0	0	0	2	1	5	1	10	5
	12	1	0	15	2	4	(1)	0	10	5
	13	6	2	15	1	0	1	0	5	15
	14	2	2	13	2	2	5	1	15	1
	15	0	0	12	1	2	6	1	-	-
	16	0	1	7	1	1	(5)	1	-	-
	17	0	1	12	1	2	(5)	1	-	-
	18	0	0	6	1	2	(5)	1	-	-
	19	5	26	36	0	30	2	0	-	-
	20	3	14	7	8	8	12	1	2	2
	21	(1)	0	2	1	1	7	1	-	-
	22	0	0	2	4	2	7	1	2	2
	23	0	0	2	4	2	7	1	2	2
	24	0	0	2	4	2	7	1	2	2
	25	0	0	2	4	2	7	1	2	2
	26	16	0	10	2	2	2	0	10	20
	27	16	8	12	5	4	2	0	10	25

Variable Number →		% Progrs Type I 52	% Progrs Type II 53	% Progrs Type IIII 54	% Progrs Type IV 55	Type I Prog Exp. 56	Type II Prog Exp 57	Type IIII Prog Exp 58	Type IV Prog Exp 59
Data Point	1	0	33	67	0	0.0	1.0	4.0	0.0
	2	14	29	38	19	0.0	4.0	7.0	8.5
	3	0	33	50	17	0.0	3.0	7.5	11.5
	4	40	44	10	6	1.5	5.5	6.5	6.5
	5	6	14	37	43	5.0	5.0	5.0	5.0
	6	25	59	15	1	3.0	6.0	15.0	15.0
	7	28	40	24	8	0.0	2.0	5.5	7.0
	8	16	39	39	6	0.0	2.0	6.0	7.5
	9	9	41	41	9	0.0	2.5	6.5	8.0
	10	33	34	33	0	0.0	0.0	0.0	0.0
	11	0	70	26	4	0.0	2.5	4.0	5.0
	12	0	50	30	20	0.0	3.0	4.0	5.0
	13	0	90	10	0	0.0	0.0	0.0	0.0
	14	17	83	00	0	1.3	6.0	0.0	0.0
	15	25	38	25	12	2.0	3.0	4.0	5.0
	16	0	62	25	13	0.0	4.0	4.0	4.0
	17	18	36	27	19	1.0	2.0	5.0	6.0
	18	0	67	33	0	0.0	3.5	15.0	0.0
	19	0	87	13	0	0.0	8.0	12.0	0.0
	20	15	70	15	0	0.0	3.0	4.0	0.0
	21	8	50	34	8	0.0	0.0	0.0	0.0
	22	12	50	25	13	2.3	2.3	2.3	2.3
	23	12	50	25	13	2.3	2.3	2.3	2.3
	24	26	59	8	7	0.0	0.0	0.0	3.0
	25	26	59	8	7	0.0	0.0	0.0	3.0
	26	13	37	37	13	2.0	3.0	4.0	5.0
	27	12	35	41	12	2.0	3.0	4.0	5.0

Variable Number →		% Progrs in Ops Desgn 60	% Progrs in Ops Desgn in Prog Desgn 61	% Prgrs in Prog Desgn 62	% Progrs Remain Entire Job 63	No. Terminations per mo. 64	No. Hires per mo. 65
Data Point	1	33	100	100	33	0	0
	2	60	75	70	45	0	5
	3	17	100	67	50	0	2
	4	29	65	88	72	3	4
	5	100	100	100	60	2	4
	6	7	67	20	55	2	5
	7	0	0	52	84	2	2
	8	0	0	45	61	3	2
	9	7	100	47	47	3	1
	10	33	100	100	17	(0)	(0)
	11	0	0	0	83	1	6
	12	30	100	60	100	2	8
	13	10	100	50	50	1	0
	14	15	100	100	5	1	0
	15	50	100	100	38	0	0
	16	25	100	63	25	0	0
	17	45	100	64	73	0	0
	18	78	100	100	56	0	0
	19	18	100	82	73	1	0
	20	80	100	100	57	0	5
	21	0	0	83	50	0	0
	22	38	100	100	100	0	2
	23	50	100	100	100	1	2
	24	0	0	100	83	1	2
	25	0	0	100	67	1	2
	26	0	0	50	100	0	0
	27	35	67	35	94	0	0

Variable Number →		Project or Funct. Org. X	Management Documents: System Design 66	Program Design 67	Error Detection X	Computer Use 68	Unavail. Computer 69
Data Point	1	2	0	0	1	0	0
	2	1	1	1	1	0	0
	3	2	1	1	1	0	0
	4	2	1	1	1	1	1
	5	2	1	1	1	0	0
	6	2	1	1	1	0	0
	7	2	0	0	1	0	0
	8	2	0	0	1	0	0
	9	2	0	0	1	0	0
	10	2	1	0	1	0	0
	11	2	0	0	1	0	1
	12	2	1	1	1	0	1
	13	1	0	0	1	1	0
	14	2	1	0	1	0	0
	15	2	1	1	0	0	0
	16	2	0	0	0	0	0
	17	2	0	1	0	1	1
	18	2	1	1	1	0	1
	19	1	1	0	1	0	1
	20	2	0	1	1	0	0
	21	2	1	0	0	1	1
	22	2	1	1	0	1	0
	23	2	1	1	0	1	0
	24	2	1	1	1	1	0
	25	2	1	1	1	1	0
	26	2	1	1	1	0	0
	27	2	1	1	1	0	0

Variable Number →		Management Documents:	Communic. with Agencies 70	Con-currence 71	Cost Control 72	Management Control 73	Document Control 74	Standards 75
Data Point	1		0	0	0	0	0	1
	2		0	1	0	0	1	0
	3		1	1	1	0	0	0
	4		0	1	1	1	1	1
	5		0	0	0	1	1	1
	6		1	0	0	1	1	1
	7		0	1	0	0	0	0
	8		0	1	0	1	1	1
	9		0	1	0	1	1	1
	10		0	0	0	0	1	0
	11		0	1	1	0	1	0
	12		0	1	1	1	1	1
	13		0	0	1	1	1	1
	14		0	0	0	0	1	0
	15		1	1	1	1	1	1
	16		1	1	1	0	1	0
	17		1	1	1	1	1	1
	18		1	1	0	0	1	1
	19		0	0	0	1	0	1
	20		0	0	0	1	1	1
	21		1	1	0	0	0	0
	22		1	1	1	1	0	1
	23		1	1	1	1	0	1
	24		1	0	1	1	0	1
	25		1	0	1	1	0	1
	26		1	1	0	1	0	0
	27		1	1	0	1	0	0

Variable Number →		No. Agencies Concurrence 76	No. Agencies Experienced 77	No. Agencies Decision-Making 78	Schedule Slipped 79	Computer Oper by Another Agency 80	Prog Developed Non-Operational Site 81	Prog Devel. Several Sites 82	Trip Mileage (Thous.) 83
Data Point	1	0	0	0	0	1	0	0	7
	2	3	0	0	1	1	0	0	189
	3	3	0	0	0	1	0	0	29
	4	1	0	0	0	1	1	1	544
	5	3	1	0	0	1	1	0	0
	6	8	3	6	0	1	1	0	540
	7	0	0	0	0	1	1	1	75
	8	0	0	0	0	1	1	1	45
	9	0	0	0	0	1	1	1	60
	10	1	0	0	1	1	0	1	4
	11	0	0	0	1	1	1	1	12
	12	1	1	0	1	1	1	1	0
	13	1	1	0	1	1	1	0	198
	14	1	0	0	1	1	1	0	10
	15	1	1	1	0	1	1	1	132
	16	1	1	1	0	1	1	1	117
	17	1	1	1	0	1	1	1	0
	18	5	4	4	0	1	0	0	0
	19	4	4	4	0	1	1	1	28
	20	3	3	3	0	0	1	1	306
	21	0	0	0	0	0	1	1	63
	22	5	5	5	0	0	1	1	15
	23	4	4	4	0	0	1	1	15
	24	5	5	5	0	0	1	0	12
	25	4	4	4	0	0	1	0	12
	26	2	0	0	1	1	1	1	8
	27	2	0	0	1	1	1	1	5

Variable Number →		Man Mos. Prog Desgn Code Test 84	Other Man Mo. 85	Averg No. Progrs Particip. 86	Months Elapsed 87	Total Computg Hours 88	Man Mos. Travel 89	Number Delivered Instruct. (Thousands) 90	Man Mos. Prog Desgn Change 91
Data Point	1	20	5	3	17	(500)	1	54	1
	2	620	72	19	24	1930	40	86	60
	3	46	15	6	13	275	4	93	2
	4	1473	193	68	29	9026	(18)	320	1100
	5	1653	234	35	56	7128	0	300	500
	6	1446	360	80	18	8426	15	277	100
	7	183	47	25	11	581	6	35	40
	8	256	66	31	16	1291	3	39	41
	9	300	77	36	13	1095	4	47	48
	10	96	13	6	25	300	3	17	6
	11	183	10	11	14	195	3	58	0
	12	143	21	10	19	130	0	30	12
	13	201	40	20	15	320	(2)	37	32
	14	232	33	14	20	850	5	46	45
	15	185	14	6	22	427	(0)	25	40
	16	64	6	8	8	633	(0)	14	(6)
	17	79	8	11	5	600	0	45	(8)
	18	27	9	6	4	222	0	17	4
	19	604	120	45	18	2249	8	110	60
	20	873	160	40	24	5667	11	55	200
	21	260	67	12	21	1625	8	50	10
	22	121	27	7	18	443	1	22	3
	23	121	27	7	18	443	1	25	2
	24	53	14	6	21	274	1	8	3
	25	53	14	6	21	274	1	10	2
	26	45	6	8	5	210	17	8	2
	27	110	12	17	5	600	8	22	6

Variable Number →		Computer Hrs Prog Design Change 92	Document Pages Design Change 93	Man Mos Eval Design Change 94	No. pgs Internal Documents (Hundreds) 95	No. Pages External Documents (Hundreds) 96	No. Other Personnel 97	Man Mos. Concurrence 98
Data Point	1	0	5	1	1	2	0	0
	2	300	500	18	5	50	18	24
	3	20	25	1	1	4	(0)	2
	4	250	818	20	60	140	(20)	12
	5	1282	0	20	15	15	35	5
	6	300	2000	100	75	68	24	28
	7	80	300	20	(9)	10	9	0
	8	129	325	10	10	15	11	8
	9	110	250	10	12	18	9	6
	10	50	200	4	(10)	6	0	2
	11	0	0	0	(5)	2	0	0
	12	30	(500)	(2)	(25)	(8)	2	(5)
	13	(32)	(1000)	(6)	(35)	(7)	(5)	(7)
	14	100	400	6	40	7	(4)	30
	15	15	200	(6)	(40)	(5)	4	(2)
	16	(63)	(200)	(1)	(20)	3	1	(1)
	17	(60)	(200)	(2)	(25)	(7)	1	(1)
	18	50	20	1	1	3	10	(0)
	19	170	300	20	15	10	5	8
	20	800	650	20	22	40	9	170
	21	45	125	0	7	9	0	0
	22	10	10	1	(5)	(4)	0	1
	23	11	13	1	(6)	(4)	(0)	1
	24	8	20	1	(2)	2	(0)	0
	25	8	10	1	(2)	1	(0)	0
	26	5	5	3	1	6	0	2
	27	20	15	9	1	6	14	10

APPENDIX IV--FREQUENCY COUNT OF ACCURACY RESPONSES

INTRODUCTION

In an attempt to determine the accuracy with which questions were answered in the questionnaire, the following accuracy index was devised:

TABLE FOR ACCURACY VALUES

From Records	From Memory	Judgment
1. Very accurate	4. Accurate recollection	7. Confident
2. Good estimate	5. Good guess	8. Good guess
3. Unreliable	6. Very hazy	9. Estimate

Appendix IV is a frequency count of these responses for those variables that were specifically tagged for this additional information. Numbers in parenthesis refer to question number in the event no variable number was assigned.

FREQUENCY COUNT OF ACCURACY RESPONSES

(Column numbers refer to Accuracy Index Table, page 101)

Variable Number \ Accuracy	1	2	3	4	5	6	7	8	9	Blanks	Percentage Blanks
1	9	2	0	5	4	0	1	1	0	3	12
3	9	3	0	6	4	0	0	1	0	2	8
4	9	2	1	6	5	0	0	0	0	3	12
5	6	3	0	5	7	0	1	0	0	3	12
6	2	1	0	0	5	2	1	0	4	10	40
7	5	4	0	5	2	0	1	1	1	6	24
8	12	1	0	1	3	0	1	0	1	6	24
9	17	0	0	3	2	0	0	0	0	3	12
11	4	4	0	3	3	1	0	0	2	8	32
90	9	6	0	0	3	0	1	0	0	6	24
16	4	4	0	1	3	0	0	0	1	12	48
17	5	2	0	0	3	0	0	1	1	13	52
18	4	2	0	0	2	2	0	1	1	13	52
20	9	2	0	2	0	0	0	1	0	11	44
21	4	2	0	1	2	0	0	1	1	14	56
22	5	1	0	3	3	1	0	3	0	9	36
23	2	2	0	1	2	0	0	5	3	10	40
27	3	4	0	6	6	0	2	0	0	4	16
30	2	3	0	1	7	4	0	0	2	6	24
91	0	4	0	0	6	1	0	1	4	9	36
92	0	3	1	0	7	1	0	1	3	9	36
93	0	2	0	0	4	3	0	4	3	9	36
94	0	2	0	0	4	2	1	1	6	9	36
32	20	0	0	1	0	0	0	0	0	4	16
33	7	1	0	3	3	0	0	0	0	11	44
34	8	1	0	2	4	0	0	1	0	9	36
38	7	3	0	2	2	1	0	0	0	10	40
(C-1)	18	0	0	3	0	0	0	0	0	4	16
40	1	5	0	0	7	0	0	1	2	9	36
41	3	3	0	3	6	0	4	1	0	5	20
(C-5)	1	1	0	1	1	0	1	8	2	10	40
42	4	5	0	6	0	0	3	2	0	5	20
43	3	6	0	7	3	0	0	0	0	6	24
44	10	1	0	0	0	0	0	0	0	14	56
45	5	3	0	4	2	0	1	0	0	10	40
46	9	2	0	3	0	0	0	0	0	11	44
52-59	4	7	1	1	6	1	0	0	1	4	16
60-65	1	6	1	1	4	1	0	2	3	6	24
(E-1)	12	1	0	4	2	0	0	0	0	6	24
76-78	5	1	0	3	3	0	3	1	0	9	36
98	3	2	0	1	4	1	0	0	3	11	44
80	12	0	0	6	1	0	0	0	0	6	24
81	11	0	1	6	0	0	0	0	0	7	28
Totals	264	107	5	106	135	21	21	38	44	359	32.6
Percentage	24.0	9.7	0.5	9.6	12.3	1.9	1.9	3.5	4.0		

APPENDIX V--VALIDITY TABLES

INTRODUCTION

These tables present the correlations of all predictor variables with costs. The first table presents the correlations for a sample size of N=26 and the second table presents the recomputed correlations for the analysis with all the extremely large data points removed (N=24). The reader will note a significant change in the values of the correlation coefficients.

While the cost variables are defined in the column headings, for economy of space, the machine variables are not defined in these tables. A complete definition of all variables will be found in Appendix II. These tables have also been referred to in the text as the correlation matrix. The decimal points have all been omitted, but the values are, of course, in hundredths.

VALIDITY TABLE

N = 26

Var. No.	Man Mos. Prog Desgn, Code, Test 84	Other Man Mos. 85	Average No. Progrs Particip. 86	Months Elapsed 87	Total Computg Hours 88	Man Mos. Travel 89	No. Delivered Instruct. (1000's) 90	Man Mos. Prog Desgn Change 91	Computer Hrs Prog Desgn Change 92	Document Pages Desgn Change 93	Man Mos. Eval Desgn Change 94	No. pgs Internal Documents (100's) 95	No. pgs External Documents (100's) 96	No. Other Personnel 97	Man Mos. Concurrence 98	Σ(84, 85, 89, 98) 99	Σ(91, 94) 100
1	17	18	21	14	13	05	29	15	-03	13	22	12	22	27	-24	15	17
2	-16	-11	-20	28	-16	-26	-30	-17	06	-21	-18	-23	-26	-32	16	-14	-18
3	-13	02	02	-29	-13	-19	-05	-24	-26	-03	11	-12	-21	04	-23	-12	-22
4	22	21	26	-01	18	20	17	15	23	27	24	22	21	38	18	22	17
5	-12	-10	-07	-02	-16	-11	-07	-12	-06	-03	-12	-12	-10	-12	-09	-12	-12
6	34	48	55	05	30	13	26	05	25	45	51	29	23	54	10	36	09
7	14	12	19	18	13	-02	08	15	04	27	21	33	20	35	-05	13	17
8	22	23	24	25	22	15	17	16	19	09	21	-06	24	32	17	23	18
9	-09	-04	-20	29	-02	-18	-24	-09	14	-23	-18	-29	-14	-30	26	-06	-10
10	67	60	53	35	70	43	56	52	62	52	53	46	68	56	51	68	55
11	89	86	85	32	87	48	94	63	40	69	78	69	78	69	16	87	68
12	23	17	12	35	20	24	09	13	14	18	13	11	18	10	13	22	14
13	69	68	64	62	68	35	71	58	34	52	55	49	68	55	15	68	61
14	12	11	18	-08	05	06	16	02	07	-03	06	-01	-02	07	-05	11	02
15	10	02	02	24	08	-07	01	14	11	-00	01	28	06	05	13	09	14
16	65	64	73	24	64	10	62	55	32	74	56	79	60	62	20	65	59
17	70	67	68	24	70	23	77	58	21	78	62	80	70	63	04	68	63
18	83	74	78	43	85	31	90	82	31	67	60	76	87	60	08	80	86
19	80	76	76	37	80	28	84	64	32	80	69	88	76	61	15	78	69
20	22	32	23	37	23	-13	29	12	-04	25	28	29	18	11	-13	22	14
21	80	90	85	14	79	39	83	40	36	80	90	66	67	74	12	80	47
22	23	26	24	15	25	07	22	14	20	01	19	-05	22	31	09	23	16
23	-29	-34	-28	-55	-25	-10	-27	-09	-25	-26	-26	-03	-17	-07	-19	-31	-11
24	-22	-21	-19	-05	-24	-33	-07	-12	-35	-07	-20	05	-25	-35	-25	-23	-14
25	26	25	19	28	21	44	24	10	34	16	22	-10	27	27	23	27	12
26	36	40	35	53	36	09	20	15	38	25	33	11	24	23	30	38	17
27	36	32	28	28	35	26	30	23	31	29	25	22	29	24	29	37	25
28	40	38	43	16	36	24	26	28	45	56	34	43	41	55	31	41	31
29	20	31	30	10	14	16	09	-12	35	31	32	03	05	20	23	23	-08
30	78	71	74	42	82	31	58	63	83	63	52	59	73	62	74	80	67
31	26	20	28	25	22	29	24	27	25	20	21	21	33	35	10	25	29
32	-30	-22	-20	-09	-27	-19	-45	-23	-07	-29	-25	-48	-30	-22	07	-28	-25
33	56	52	65	18	48	34	53	42	45	56	48	49	53	64	23	56	45
34	23	17	12	12	19	31	24	15	-11	-03	16	06	21	08	-20	20	16
35	-06	-11	-04	-20	-14	02	06	02	-37	03	05	15	00	04	-42	-09	02
36	09	03	-05	33	05	22	14	13	-18	19	11	23	18	08	-14	07	14
37	34	25	21	14	26	25	45	28	04	43	26	41	38	38	-08	31	29
38	45	41	49	-20	45	32	44	37	20	29	33	30	38	37	13	44	39
39	78	81	78	39	73	43	74	46	42	74	80	63	71	75	17	77	52
40	88	75	87	34	86	53	89	86	45	62	56	63	94	78	15	85	89
41	-32	-28	-30	-22	-31	00	-33	-32	-04	-51	-35	-68	-32	-27	04	-30	-34
42	-43	-39	-30	-62	-47	-05	-32	-37	-36	-35	-33	-52	-38	-19	-37	-44	-39

VALIDITY TABLE

N = 26

Var. No.	Man Mos. Prog Desgn, Code, Test (84)	Other Man Mos. (85)	Average No. Progrs Particip. (86)	Months Elapsed (87)	Total Computg Hours (88)	Man Mos. Travel (89)	No. Delivered Instruct. (1000's) (90)	Man Mos. Prog Desgn Change (91)	Computer Hrs Prog Desgn Change (92)	Document Pages Desgn Change (93)	Man Mos. Eval Desgn Change (94)	No. pgs Internal Documents (100's) (95)	No. pgs External Documents (100's) (96)	No. Other Personnel (97)	Man Mos. Concurrence (98)	Σ(84, 85, 89, 98) (99)	Σ(91, 94) (100)
43	15	14	12	-04	21	-07	07	16	09	09	12	29	14	04	15	15	17
44	41	45	55	05	39	04	46	35	16	41	42	35	43	57	-07	40	38
45	-08	-09	01	-37	-11	28	-15	-11	03	-02	-01	-16	-12	08	12	-07	-11
46	22	24	38	-00	16	08	05	01	44	05	14	-05	-01	16	40	25	02
47	-00	02	23	-10	-09	10	-01	-01	05	00	03	-09	-01	22	-08	-01	-01
48	-06	05	12	-07	-03	-17	-28	-14	27	-07	04	-21	-12	11	36	-02	-13
49	12	18	34	-03	04	00	01	-07	22	-01	14	-11	-08	11	11	13	-06
50	15	15	-00	42	24	-03	-04	08	50	-09	01	-07	10	-07	56	18	08
51	14	14	05	15	20	-01	17	13	17	-03	08	06	18	10	11	14	14
52	40	33	34	46	43	22	34	48	19	24	31	38	50	34	08	39	50
53	13	18	18	06	09	-22	02	-03	18	26	09	26	-09	-07	25	15	-02
54	-34	-29	-30	-32	-33	02	-34	-36	-20	-32	-18	-45	-29	-07	-19	-34	-36
55	-26	-33	-33	-22	-28	15	-18	-13	-26	-28	-26	-21	-09	-14	-27	-28	-15
56	41	45	33	21	41	02	44	23	03	41	48	60	33	26	-02	40	27
57	59	54	58	17	52	28	59	36	38	43	47	54	45	46	19	58	39
58	03	05	12	-57	04	30	04	-03	-04	-04	17	-13	03	29	-04	03	-01
59	38	44	44	-12	35	35	49	13	07	42	57	31	38	52	-12	37	18
60	08	-03	-10	05	10	13	-06	09	42	-05	-09	-00	12	17	45	10	08
61	06	03	01	03	04	-13	06	02	17	15	01	26	02	03	19	06	02
62	-14	-22	-31	44	-08	-16	-19	11	07	-32	-36	-13	-05	-31	14	-14	08
63	-04	-04	01	-23	-04	-02	-04	04	-14	-16	-05	-24	-01	-05	-12	-05	03
64	43	44	61	22	39	01	45	44	08	41	34	40	45	44	-09	41	46
65	46	40	34	38	40	33	37	26	41	39	34	23	42	32	33	46	28
66	11	08	-04	29	08	24	17	10	-17	-02	09	05	13	07	-20	09	11
67	18	12	02	02	22	23	16	18	17	06	14	08	26	20	17	18	19
68	-01	-06	-08	22	04	-17	05	24	-20	-06	-21	04	11	-23	-20	-03	21
69	15	05	12	-05	13	-06	25	29	-06	-04	-12	10	18	-02	-16	12	27
70	-25	-14	-29	-38	-16	-22	-17	-27	-34	-22	-02	-13	-25	-21	-26	-25	-26
71	-18	-27	-18	-39	-19	08	-08	07	-27	-34	-28	-22	02	04	-33	-21	04
72	-17	-27	-27	08	-16	-39	-03	13	-34	-13	-32	10	-04	-39	-27	-21	10
73	26	28	35	07	26	-08	14	20	15	25	20	29	19	16	16	26	21
74	34	26	29	13	29	11	23	24	36	47	22	55	34	38	27	33	25
75	24	28	28	20	25	-30	18	20	16	25	16	29	18	12	13	24	21
76	30	43	25	06	30	15	25	-07	20	32	51	14	13	26	14	33	-02
77	03	14	01	10	05	-24	-08	-13	10	00	11	-06	-12	-11	13	05	-12
78	21	36	20	10	23	-14	11	-10	17	19	38	10	00	07	15	24	-07
79	-16	-25	-22	-04	-26	28	-20	-16	-14	03	-14	-05	-12	-02	-05	-17	-17
80	07	03	18	-35	01	15	23	09	-15	22	17	27	13	33	-27	05	10
81	17	20	29	-02	17	-20	04	13	06	18	13	34	06	00	10	18	14
82	-15	-16	-15	-01	-08	-17	-05	-01	-11	-23	-17	-19	-10	-21	-06	-15	-03
83	92	86	83	38	93	52	84	71	61	82	74	78	88	76	41	92	76

VALIDITY TABLE

N = 24

Var. No.	Man Mos. Prog Desgn, Code, Test	Other Man Mos.	Average No. Progrs Particip.	Months Elapsed	Total Computg Hours	Man Mos. Travel	No. Delivered Instruct. (1000's)	Man Mos. Prog Desgn Change	Computer Hrs Prog Desgn Change	Document Pages Desgn Change	Man Mos. Eval Desgn Change	No. pgs Internal Documents (100's)	No. pgs External Documents (100's)	No Other Personnel	Man Mos Concurrence	Σ(84, 85, 89, 98)	Σ(91, 94)
	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	-05	-02	05	08	-15	-03	24	-09	-11	-08	16	-10	07	16	-26	-07	-05
2	13	24	03	41	18	-19	-16	14	16	-03	06	-04	-08	-19	19	15	13
3	-22	-10	02	-25	-23	-19	-03	-20	-29	-23	-10	-22	-28	04	-26	-22	-19
4	19	19	25	-06	13	17	11	27	20	31	40	18	24	40	17	20	30
5	01	06	09	03	-06	-07	20	-03	-01	14	-03	00	07	-03	-08	01	-03
6	25	38	59	04	17	06	11	28	18	27	44	09	29	50	06	25	31
7	-05	-08	06	13	-09	-10	-29	05	-02	24	21	29	08	31	-07	-06	07
8	11	15	15	21	12	09	-03	12	14	-09	20	-34	21	27	16	12	14
9	09	19	-13	38	26	-13	-29	12	20	-21	-21	-27	-00	-28	28	12	08
10	36	26	08	21	44	29	-04	49	57	16	26	-02	51	24	59	39	47
11	55	48	47	13	42	39	73	33	28	03	51	-01	26	24	21	52	37
12	20	09	-01	33	15	20	-14	09	10	12	07	-01	13	-00	12	18	09
13	29	39	22	59	23	16	28	29	19	12	34	-05	27	18	15	30	31
14	29	31	37	-09	17	08	56	10	08	01	32	03	00	14	-04	26	14
15	18	11	05	23	14	-08	-03	29	12	07	24	46	03	08	14	17	29
16	22	27	45	04	15	-20	-02	32	16	63	34	60	05	30	22	23	33
17	-04	-08	01	-02	-16	-12	-02	-07	-09	51	-07	45	-03	15	-04	-05	-07
18	21	22	26	34	13	-11	13	16	14	35	24	34	-00	-13	10	20	18
19	14	10	14	23	06	-07	07	20	11	53	08	67	01	-00	18	14	19
20	-11	01	-09	35	-11	-27	-00	-09	-17	-10	-17	04	-13	-19	-18	-11	-11
21	37	48	56	-14	29	30	44	26	25	17	44	-15	37	45	09	36	29
22	18	28	19	11	23	02	21	16	16	-19	23	-28	22	29	08	19	17
23	-30	-35	-23	-27	-23	-05	-36	-19	-21	-18	-22	19	-17	07	-18	-30	-20
24	-22	-21	-15	-02	-28	-31	13	-26	-33	06	-27	24	-40	-37	-24	-23	-27
25	32	29	17	28	24	43	39	26	32	09	32	-32	49	26	22	32	28
26	34	40	27	54	36	02	-02	32	33	08	30	-13	22	09	29	35	32
27	27	21	11	22	25	19	16	21	25	17	16	03	17	08	28	27	21
28	31	30	36	08	25	16	-04	41	40	66	40	35	45	51	31	32	42
29	42	49	50	19	34	20	33	37	37	39	49	00	41	26	22	42	40
30	67	69	60	28	79	13	08	87	86	54	54	29	61	39	89	71	85
31	20	19	26	18	12	24	16	28	21	22	45	14	33	33	10	20	31
32	-03	12	13	02	05	-08	-44	03	03	-09	-01	-38	-05	01	10	00	03
33	45	47	61	05	29	23	45	49	38	56	68	32	47	56	23	44	53
34	-00	-11	-18	05	-10	25	-02	-24	-21	-42	-12	-24	-06	-15	-22	-04	-23
35	-32	-44	-25	-27	-54	-02	-17	-40	-43	-10	-12	09	-27	-07	-43	-36	-37
36	-25	-37	-45	29	-37	14	-28	-26	-28	01	-24	07	-10	-13	-16	-25	-26
37	-01	-18	-21	01	-21	13	24	-08	-10	26	-11	18	10	16	-11	-04	-08
38	19	19	30	-42	19	20	14	09	09	02	19	-03	-02	14	12	19	11
39	37	37	40	37	21	30	17	36	31	27	49	04	53	46	17	36	39
40	69	72	83	01	54	53	55	53	50	50	77	04	68	71	27	67	58
41	12	17	09	-10	18	18	23	-02	11	-38	-10	-62	16	05	08	13	-04
42	-13	-07	10	-59	-21	12	28	-32	-27	-06	-03	-32	-03	16	-38	-15	-29

VALIDITY TABLE

N = 24

Var. No.	Man Mos. Prog Desgn, Code, Test	Other Man Mos.	Average No. Progrs Particip.	Months Elapsed	Total Computg Hours	Man Mos. Travel	No. Delivered Instruct. (1000's)	Man Mos. Prog Desgn Change	Computer Hrs Prog Desgn Change	Document Pages Desgn Change	Man Mos. Eval Desgn Change	No. pgs Internal Documents (100's)	No. pgs External Documents (100's)	No. Other Personnel	Man Mos. Concurrence	Σ(84, 85, 89, 98)	Σ(91, 94)
	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
43	-06	-08	-09	-12	06	-16	-36	04	03	-12	-10	21	-12	-15	14	-04	02
44	05	19	36	-11	-03	-14	13	11	03	15	29	00	11	40	-11	05	14
45	16	14	29	-34	11	38	06	14	09	22	39	-03	10	29	13	17	18
46	68	76	86	05	63	15	54	59	52	28	75	10	32	37	42	68	63
47	28	40	67	-08	13	17	43	24	12	31	65	08	28	50	-07	27	31
48	22	38	48	02	33	-11	-26	41	35	07	39	-12	23	33	37	26	42
49	51	64	82	04	39	06	49	39	28	16	70	01	20	31	12	49	44
50	31	33	01	45	53	-03	-08	46	53	-15	05	-10	29	-09	56	35	41
51	01	02	-13	11	12	-07	05	09	13	-27	-11	-09	12	-01	10	02	07
52	-02	-04	-07	34	01	06	-41	10	06	-09	18	05	08	03	07	-01	12
53	27	30	29	10	22	-23	10	26	18	40	08	40	-11	-11	25	27	24
54	-13	-14	-08	-22	-09	16	-08	-19	-12	-25	-05	-37	10	22	-19	-14	-17
55	-19	-28	-28	-21	-25	23	-03	-28	-21	-17	-17	-08	10	-00	-26	-21	-28
56	-15	-19	-29	12	-17	-22	-21	-11	-18	-19	-22	30	-20	-24	-09	-16	-13
57	47	39	42	05	31	15	58	34	29	18	42	35	24	23	18	43	36
58	-10	-13	06	-63	-07	30	-12	-11	-08	-23	12	-34	-02	31	-05	-09	-08
59	-10	-14	03	-27	-20	26	20	-13	-12	-13	11	-19	19	26	-22	-11	-10
60	32	17	-03	05	39	17	05	39	48	12	17	13	41	34	47	33	37
61	14	08	03	04	12	-13	23	19	18	29	04	43	10	06	19	14	18
62	04	08	-24	49	17	-13	-15	14	16	-10	-09	09	-02	-23	20	06	11
63	-11	-06	01	-28	-13	-03	-18	-18	-15	-25	-04	-39	-14	-08	-12	-10	-16
64	07	23	48	05	-07	-19	01	07	-06	24	24	10	02	21	-13	06	10
65	34	22	11	33	23	24	18	28	34	22	15	-06	42	13	33	34	27
66	-14	-20	-37	25	-22	18	-09	-29	-25	-33	-20	-19	-14	-11	-22	-15	-28
67	-07	-17	-33	-07	00	15	-22	00	10	-23	-15	-18	12	04	16	-05	-02
68	-24	-14	-28	13	-18	-25	-28	-28	-25	-06	-41	-04	-26	-42	-19	-23	-30
69	04	04	04	-20	-02	-15	26	-15	-10	-07	-16	02	-15	-14	-16	01	-16
70	-51	-48	-57	-36	-33	-23	-43	-46	-38	-61	-56	-26	-44	-34	-30	-50	-49
71	-22	-24	-15	-49	-26	11	-06	-29	-25	-30	-14	-19	01	19	-32	-23	-28
72	-40	-45	-48	02	-42	-45	-22	-36	-37	-08	-55	19	-42	-55	-27	-41	-40
73	12	20	29	00	12	-17	-23	18	09	15	10	18	-00	01	14	13	18
74	25	11	16	06	16	02	-00	32	30	54	10	59	34	30	26	24	30
75	08	19	17	15	12	-42	-10	18	10	15	-03	18	-06	-06	11	09	15
76	06	05	-12	08	07	08	-07	01	10	-24	-04	-34	01	-02	11	07	00
77	05	12	-03	16	11	-26	-20	06	09	-18	-08	-15	-14	-20	12	06	04
78	06	12	-03	15	13	-23	-19	06	10	-27	-07	-21	-13	-19	12	07	04
79	00	-20	-11	02	-19	38	-05	-10	-09	31	00	16	12	15	-03	-02	-09
80	-12	-22	10	-43	-29	10	25	-14	-21	19	17	24	-02	31	-29	-16	-10
81	10	19	31	-07	10	-27	-27	15	01	15	08	37	-15	-13	09	11	14
82	-14	-11	-12	-00	-01	-16	08	-09	-08	-22	-10	-17	-14	-20	-05	-13	-09
83	68	59	44	23	69	47	17	77	74	67	56	34	70	45	71	70	76

APPENDIX VI

FACTOR ANALYSIS OF PREDICTOR VARIABLES

Factor Loadings for First Six Computed Factors
After Rotation by Varimax Method (N=26)

Var. No.	Short Description	FACTOR COEFFICIENTS I	II	III	IV	V	VI
1	New System	19	34	-20	-35	21	-44
2	New Hardware	-20	-05	-24	59	27	03
3	Partic. in Reqt. Anal.	-14	22	20	-15	50	-13
4	Partic. in Oper. Design	25	48	15	-12	-01	-44
5	How Well Reqts. Known	-04	08	-45	02	02	27
6	No. System Changes	42	66	-20	-14	-01	27
7	Time of Peak Changes	32	15	-32	10	-05	-09
8	No. Commands	28	03	-38	-14	47	37
9	No. ADP Centers	-20	-39	-47	41	50	26
10	Complexity	64	-07	-31	-03	09	-14
11	Estimated Instructions	79	04	30	-01	25	-16
12	% New Instructions	08	-28	24	25	09	20
13	Table, Constants, Words	67	18	-27	05	31	-21
14	% Subroutine Instructions	-09	38	45	36	28	-46
15	% Discarded Instructions	08	-04	-02	-03	03	-34
16	Data Base Words	82	15	16	21	-09	13
17	Data Base Classes	88	-16	15	02	-13	00
18	No. Input Messages	88	-07	09	11	02	-11
19	No. Output Messages	92	-08	12	16	-10	-09
20	% Instruction Innovation	30	01	-24	-24	13	-18
21	No. Subprograms	82	13	03	-17	15	08
22	Maintainability	07	16	-26	-26	70	-08
23	% Clerical Instructions	-23	-14	35	-47	-32	-06
24	% Data Reduction Instructions	-16	-20	23	37	01	05
25	% Prediction Instructions	21	14	-43	-26	36	-02
26	% Decision Instructions	27	23	-33	34	09	-04
27	Insufficient Memory	36	-19	01	37	37	-08
28	Insufficient I/O	52	42	-18	06	-43	-10
29	Timing Constraint	16	53	-28	02	16	-06
30	No. Program Changes	73	25	-16	14	-03	11
31	Time - Peak Program Changes	33	52	-16	-28	-18	-37
32	Language Type	-29	03	-21	12	09	69
33	No. Program Tools	61	60	-04	-17	00	-27
34	Parameter Test Reqts.	11	-53	26	-07	48	18
35	Assembly Test Reqts.	20	-30	29	-15	06	-18

NOTE: Decimal points are omitted before each coefficient.

Var. No.	Short Description	FACTOR COEFFICIENTS I	II	III	IV	V	VI
36	Reqts. Specify I/O	33	-54	09	-05	15	-16
37	Reqts. for Stop Testing	42	-22	16	-31	-12	-38
38	No. Internal Documents	30	00	83	07	15	10
39	No. External Documents	86	19	-21	-25	08	06
40	Computer Hours/Week	82	20	11	-04	08	-04
41	Computer Adeq. Parameter Test	-57	04	04	-17	48	19
42	Computer Operation Documented	-54	28	31	-40	01	12
43	Computer Design Interrupted	23	-36	03	-12	-03	48
44	Core Size	49	53	-26	-22	-04	-04
45	No. EDP Components	-14	17	70	03	01	20
46	No. Displays	-03	67	34	37	32	04
47	I/O Equipments	-01	89	06	-25	-07	03
48	Input Equipments	-03	45	-40	-06	16	61
49	Output Equipments	-07	81	23	19	31	07
50	Pieces EAM Equipment	01	-14	-79	23	20	03
51	EAM Adequate	11	-10	-71	-17	09	-01
52	% Type I Programmers	54	-18	-33	00	05	12
53	% Type II Programmers	13	02	22	78	-03	09
54	% Type III Programmers	-49	27	02	-39	-12	-02
55	% Type IV Programmers	-20	-32	03	-69	16	01
56	Type I Programmer Experience	51	-38	-11	-05	33	-15
57	Type II Programmer Experience	51	28	22	12	13	-42
58	Type III Programmer Experience	-03	06	71	-36	19	08
59	Type IV Programmer Experience	50	08	-02	-74	14	07
60	% Programmers in Ops Design	-08	-10	-15	04	17	-59
61	% Programmers in Ops and Prg Design	05	07	00	14	01	-70
62	% Programmers in Program Design	-21	-19	-41	47	23	-48
63	% Programmers in Whole Job	-08	-17	28	-26	55	48
64	No. Terminations per Month	58	42	-07	-13	-01	41
65	No. Hires per Month	45	-09	-12	-20	08	18
66	System Design Documented	14	-43	22	04	40	-35
67	Program Design Documented	19	-53	01	-29	57	-19
68	Computer Use Documented	03	-55	-23	25	28	29
69	Unavailable Computer Documented	01	-12	37	05	-04	-03
70	Communicating Agency Documented	-20	-60	02	-21	43	-05
71	Concurring Agency Documented	-16	-10	07	-78	-02	09
72	Cost Control Documented	02	-62	-14	-07	06	03
73	Management Control Documented	28	-05	19	04	54	31
74	Document Control Documented	45	07	-10	-03	-52	-20
75	Standards Documented	21	08	-18	21	49	-07
76	No. Concurring Agencies	27	-22	-01	19	71	-23
77	No. Experienced Agencies	-03	-19	-17	46	73	-07
78	No. Decision Agencies	15	-17	-17	38	73	-08
79	Schedule Slipped	-04	-17	46	00	-50	08
80	Computer Operated by Other	19	38	48	-39	-50	-32
81	Program Developed at Other Site	25	-12	19	17	13	68
82	Program Developed at Several Sites	-32	21	-06	-02	-12	-12
83	Trip Mileage	88	-03	02	-01	04	-05

SUMMARY OF REGRESSION ANALYSES

APPENDIX VII--SUMMARY OF CORRELATION AND REGRESSION ANALYSES

INTRODUCTION

The following eight tables are summaries of the results of the correlation and regression analyses. Each potential predictor variable considered in the final regression analysis is listed with a short description of the variable (a more complete description will be found in Appendix II). Various statistical relationships of the variables are presented as well as the final regression equation.

TABLE 1

SUMMARY OF CORRELATION AND REGRESSION ANALYSIS FOR COST VARIABLE 84

Number of Man Months to Design, Code and Test

Variable Number	11	10	39	38	16	26	64
Short Description	Est. Instr. (1000's)	Complex-ity	Ext. Docts.	Int. Docts.	D/B Words (Log_{10})	% Decis. Instr.	Terms. Per Month
Means	54.8	3.2	5.6	3.8	3.0	28.1	.8
Standard Deviations	66.2	.9	4.0	3.6	2.2	18.8	1.0
Validity Coefficients	.89	.67	.78	.45	.65	.36	.43
Intercorrelations Variable Number							
11	1.00	.50	.67	.59	.58	.24	.32
10	.50	1.00	.55	.05	.33	.42	.09
39	.67	.55	1.00	.06	.56	.35	.66
38	.59	.05	.06	1.00	.35	-.06	.04
16	.58	.33	.56	.35	1.00	.11	.63
26	.24	.42	.35	-.06	.11	1.00	.07
64	.32	.09	.66	.04	.63	.07	1.00
Standardized Regression Coefficients (7 variables)	.46	.25	.21	.12	.11	.07	.05
Standardized Regression Coefficients (5 variables)	.45	.26	.26	.11	.12	not se-lected	not se-lected

Multiple Correlation Coefficient	.95	Number of Data Points	26
Mean of Cost Variable	300	Standard Error of Estimate	138
Standard Error of Prediction at the Mean	141	Standard Deviation of Cost Variable	397

95% Confidence Limits at the Mean* $\pm$295 Man Months

PREDICTION EQUATION: $Y_{84} = 2.7X_{11} + 121X_{10} + 26X_{39} + 12X_{38} + 22X_{16} - 497$

*These limits will expand as predictions deviate from the mean.

TABLE 2

SUMMARY OF CORRELATION AND REGRESSION ANALYSIS FOR COST VARIABLE 87

Months Elapsed

Variable Number	13	44	26	39	11	64	10	16
Short Description	Wds. in Tbls. & Consts. (Log_{10})	Core Size (K)	% Decis. Instr.	Ext. Doc. Types	Est. Inst. (thous.)	Term. Per Mo.	Complex-ity	D/B Wds. (Log_{10})
Means	3.8	35.9	28.1	5.6	54.8	.8	3.2	3.0
Standard Deviations	1.6	19.0	18.8	4.0	66.2	1.0	.9	2.2
Validity Coefficients	.62	.05	.53	.39	.32	.22	.35	.24
Intercorrelations Variable Number								
13	1.00	.49	.38	.66	.66	.44	.53	.48
44	.49	1.00	.17	.67	.31	.70	.23	.37
26	.38	.17	1.00	.35	.24	.07	.42	.11
39	.66	.67	.35	1.00	.67	.66	.55	.56
11	.66	.31	.24	.67	1.00	.32	.50	.58
64	.44	.70	.07	.66	.32	1.00	.09	.63
10	.53	.23	.42	.55	.50	.09	1.00	.33
16	.48	.37	.11	.56	.58	.63	.33	1.00
Standardized Regression Coefficients (8 variables)	.74	-.61	.34	.29	-.27	.23	-.07	-.06
Standardized Regression Coefficients (4 variables)	.59	-.40	.32	.16	not se-lected	not se-lected	not se-lected	not se-lected

Multiple Correlation Coefficient	4.9	Number of Data Points	26
Mean of Cost Variable	16.3	Standard Error of Estimate	4.8
Standard Error of Prediction at the Mean	4.9	Standard Deviation of Cost Variable	6.8

95% Confidence Limits at the Mean* ±10.2 Months

PREDICTION EQUATION: $Y_{87} = 2.5X_{13} - .14X_{44} + .11X_{26} + .3X_{39} + 7.0$

*These limits will expand as predictions deviate from the mean.

TABLE 3

SUMMARY OF CORRELATION AND REGRESSION ANALYSIS FOR COST VARIABLE 88

Computer Hours Used

Variable Number	11	10	16	64	38	26
Short Description	Est. Instr. (1000's)	Complex-ity	D/B Wds. (Log_{10})	Terms. Per Month	Int. Doct. Types	% Decis. Instr.
Means	54.8	3.2	3.0	.8	3.8	28.1
Standard Deviations	66.2	.9	2.2	1.0	3.6	18.8
Validity Coefficients	.87	.70	.64	.39	.45	.36
Intercorrelations Variable Number						
11	1.00	.50	.58	.32	.59	.24
10	.50	1.00	.33	.09	.05	.42
16	.58	.33	1.00	.63	.35	.11
64	.32	.09	.63	1.00	.04	.07
38	.59	.05	.35	.04	1.00	-.06
26	.24	.42	.11	.07	-.06	1.00
Standardized Regression Coefficients (6 variables)	.52	.37	.11	.11	.09	.07
Standardized Regression Coefficients (3 variables)	.59	.35	.18	not se-lected	not se-lected	not se-lected

Multiple Correlation Coefficient	.94	Number of Data Points	26
Mean of Cost Variable	1482	Standard Error of Estimate	905
Standard Error of Prediction at the Mean	923	Standard Deviation of Cost Variable	2410

95% Confidence Limits at the Mean* ±1911 Hours

PREDICTION EQUATION: $Y_{88} = 21.5X_{11} + 985X_{10} + 197X_{16} - 3468$

* These limits will expand as predictions deviate from the mean.

TABLE 4

SUMMARY OF CORRELATION AND REGRESSION ANALYSIS FOR COST VARIABLE 90

Delivered Instructions (In thousands)

Variable Number	11	18	44	72	38	8
Short Description	Est. Instr. (1000's)	Input Mess. Types	Core Size (K)	Cost Contrl. Doc.	Int. Doc. Types	No. of Comds.
Means	54.8	9.0	35.9	.5	3.8	2.8
Standard Deviations	66.2	16.4	19.0	.5	3.6	1.8
Validity Coefficients	.94	.90	.46	-.03	.44	.17
Intercorrelations Variable Number						
11	1.00	.83	.31	-.14	.59	.14
18	.83	1.00	.40	.05	.41	.13
44	.31	.40	1.00	-.26	-.10	.07
72	-.14	.05	-.26	1.00	-.25	.13
38	.59	.41	-.10	-.25	1.00	-.19
8	.14	.13	.07	.13	-.19	1.00
Standardized Regression Coefficients (6 variables)	.72	.26	.14	.08	-.06	.00
Standardized Regression Coefficients (3 variables)	.63	.33	.14	not se-lected	not se-lected	not se-lected

Multiple Correlation Coefficient	.97	Number of Data Points	26
Mean of Cost Variable	59.6	Standard Error of Estimate	19.0
Standard Error of Prediction at the Mean	19.4	Standard Deviation of Cost Variable	75.2

95% Confidence Limits at the Mean* $\pm$40.2 No. Instruc. (Thous.)

PREDICTION EQUATION: $Y_{90} = .7X_{11} + 1.5X_{18} + .5X_{44} - 12.0$

*These limits will expand as predictions deviate from the mean.

TABLE 5

SUMMARY OF CORRELATION AND REGRESSION ANALYSIS FOR COST VARIABLE 96

Number of External Document Pages (In hundreds)

Variable Number	18	39	11	8	72	5
Short Description	Input Mess. Types	Ext. Doc. Types	Est. Instr. (1000's)	No. of Comds.	Cost Cont. Doc.	How well Req'ts Known
Means	9.0	5.6	54.8	2.8	.5	2.4
Standard Deviations	16.4	4.0	66.2	1.8	.5	.8
Validity Coefficients	.87	.71	.78	.24	-.04	-.10
Intercorrelations Variable Number						
18	1.00	.68	.83	.13	.05	-.17
39	.68	1.00	.67	.35	-.16	.06
11	.83	.67	1.00	.14	-.14	-.27
8	.13	.35	.14	1.00	.13	.34
72	.05	-.16	-.14	.13	1.00	.35
5	-.17	.06	-.27	.34	.35	1.00
Standardized Regression Coefficients (6 variables)	.68	.12	.13	.09	-.06	.03
Standardized Regression Coefficients (2 variables)	.72	.22	not selected	not selected	not selected	not selected

Multiple Correlation Coefficient	.89	Number of Data Points	26
Mean of Cost Variable	16.8	Standard Error of Estimate	14.4
Standard Error of Prediction at the Mean	14.7	Standard Deviation of Cost Variable	29.8
95% Confidence Limits at the Mean*	±30.4 No. pages (Hundreds)		

PREDICTION EQUATION: $Y_{96} = 1.3X_{18} + 1.7X_{39} - 4.2$

*These limits will expand as predictions deviate from the mean.

TABLE 6

SUMMARY OF CORRELATION AND REGRESSION ANALYSIS FOR COST VARIABLE 99

Sum of All Man Months

Variable Number	11	39	10	38	16	26	64
Short Description	Est. Instr. (1000's)	Ext. Doc. Types	Complex-ity	Int. Doc. Types	D/B Wds. (Log_{10})	% Decis. Instr.	Term's. Per Mo.
Means	54.8	5.6	3.2	3.8	3.0	28.1	.8
Standard Deviations	66.2	4.0	.9	3.6	2.2	18.8	1.0
Validity Coefficients	.87	.77	.68	.44	.65	.38	.41
Intercorrelations Variable Number							
11	1.00	.67	.50	.59	.58	.24	.32
39	.67	1.00	.55	.06	.56	.35	.66
10	.50	.55	1.00	.05	.33	.42	.09
38	.59	.06	.05	1.00	.35	-.06	.04
16	.58	.56	.33	.35	1.00	.11	.63
26	.24	.35	.42	-.06	.11	1.00	.07
64	.32	.66	.09	.04	.63	.07	1.00
Standardized Regression Coefficients (7 variables)	.39	.26	.26	.14	.12	.08	.00
Standardized Regression Coefficients (5 variables)	.40	.28	.28	.14	.12	not se-lected	not se-lected

Multiple Correlation Coefficient	.94	Number of Data Points	26
Mean of Cost Variable	373	Standard Error of Estimate	182
Standard Error of Prediction at the Mean	186	Standard Deviation of Cost Variable	492

95% Confidence Limits at the Mean* $\pm$389 Man Months

PREDICTION EQUATION: $Y_{99} = 3.0X_{11} + 35X_{39} + 164X_{10} + 18X_{38} + 26X_{16} - 658$

*These limits will expand as predictions deviate from the mean.

TABLE 7

SUMMARY OF CORRELATION AND REGRESSION ANALYSIS FOR COST VARIABLE 100

Man Months for Changes

Variable Number	18	11	13	38	23	10	26	8
Short Description	Input Mess. Types	Est. Instr. (1000's)	Wds. in Tbls. & Const. (Log_{10})	Int. Doc. Types	% Cler. Instr.	Complex-ity	% Decis. Instr.	No. of Comds.
Means	9.0	54.8	3.8	3.8	31.3	3.2	28.1	2.8
Standard Deviations	16.4	66.2	1.6	3.6	22.3	.9	18.8	1.8
Validity Coefficients	.86	.68	.61	.39	-.11	.55	.17	.18
Intercorrelations Variable No.								
18	1.00	.83	.66	.41	-.27	.59	.34	.13
11	.83	1.00	.66	.59	-.21	.50	.24	.14
13	.66	.66	1.00	.03	-.49	.53	.38	.41
38	.41	.59	.03	1.00	.23	.05	-.06	-.19
23	-.27	-.21	-.49	.23	1.00	-.30	-.49	-.52
10	.59	.50	.53	.05	-.30	1.00	.42	.34
26	.34	.24	.38	-.06	-.49	.42	1.00	.18
8	.13	.14	.41	-.19	-.52	.34	.18	1.00
Standardized Regression Coefficients (8 variables)	.91	-.44	.30	.23	.17	.13	-.12	.10
Standardized Regression Coefficients (3 variables)	.78	not se-lected	.19	not se-lected	.19	not se-lected	not se-lected	not se-lected

Multiple Correlation Coefficient	.94	Number of Data Points	26
Mean of Cost Variable	373	Standard Error of Estimate	182
Standard Error of Prediction at the Mean	186	Standard Deviation of Cost Variable	492

95% Confidence Limits at the Mean* ±389 Man Months

PREDICTION EQUATION: $Y_{100} = 10.4X_{18} + 27X_{13} + 1.9X_{23} - 174$

*These limits will expand as predictions deviate from the mean.

TABLE 8

SUMMARY OF CORRELATION AND REGRESSION ANALYSIS FOR COST VARIABLE 90

Number of Delivered Instructions (in thousands)

(Alternate Solution Without Using Estimated Instructions as a Predictor)

Variable Number	18	21	13	16	44	5
Short Description	Input Mess. Types	No. of Sub-Progs	Wds in Tables & Const. (Log_{10})	D/B Words (Log_{10})	Core Size (K)	How well Reqts. Known
Means	9.0	24.5	3.8	3.0	35.9	2.4
Standard Deviations	16.4	23.0	1.6	2.2	19.0	.8
Validity Coefficients	.90	.83	.71	.62	.46	-.07
Intercorrelations Variable Number						
18	1.00	.69	.66	.69	.40	-.17
21	.69	1.00	.60	.61	.55	.05
13	.66	.60	1.00	.48	.49	-.04
16	.69	.61	.48	1.00	.37	.04
44	.40	.55	.49	.37	1.00	.20
5	-.17	.05	-.04	.04	.20	1.00
Standardized Regression Coefficients (6 variables)	.64	.39	.13	-.11	-.05	.04
Standardized Regression Coefficients (3 variables)	.58	.36	.12	not se-lected	not se-lected	not se-lected

Multiple Correlation Coefficient	.88	Number of Data Points	26
Mean of Cost Variable	81	Standard Error of Estimate	113
Standard Error of Prediction at the Mean	115	Standard Deviation of Cost Variable	219

95% Confidence Limits at the Mean $\pm$238 Man Months

PREDICTION EQUATION: $Y_{90} = 2.6X_{18} + 1.2X_{21} + 5.6X_{13} - 13.9$

*These limits will expand as predictions deviate from the mean.

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*
System Development Corporation
Santa Monica, California

2a. REPORT SECURITY CLASSIFICATION
Unclassified

2b. GROUP
N/A

3. REPORT TITLE
FACTORS THAT AFFECT THE COST OF COMPUTER PROGRAMMING--A QUANTITATIVE ANALYSIS

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
Volume II

5. AUTHOR(S) *(Last name, first name, initial)*
Farr, L., Zagorski, H. J.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
September 1964	116	14

8a. CONTRACT OR GRANT NO.
AF 19(628)-3418

b. PROJECT NO.

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)
TM-1447/001/00

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
ESD-TDR-64-448, Volume II

10. AVAILABILITY/LIMITATION NOTICES
Qualified Requestors May Obtain from DDC
Available from OTS

11. SUPPLEMENTARY NOTES
None

12. SPONSORING MILITARY ACTIVITY
Directorate of Computers, Hq Electronic Systems Division, L.G. Hanscom Field, Bedford Massachusetts 01731

13. ABSTRACT

Results of an exploratory analysis aimed at deriving better cost-estimating relationships for computer programming development are presented. Based upon previous work that hypothesized an initial list of factors affecting cost, the report describes the steps taken to collect and analyze data for the purpose of supporting or rejecting the presumed factors. As a result, equations that estimate costs in terms of such resources as man months and computer hours have been derived. Since these estimating devices were evolved from a small and, perhaps, unrepresentative sample of programs, the use of these equations is not recommended for actual planning. The study concludes that multivariate regression analysis, supplemented by pertinent judgment and intuition, is an appropriate tool for deriving cost-estimating relationships. To arrive at more useful prediction equations, recommendations are made for continuing the research. These include increasing the sample size and improving the questionnaire used to collect data. The basic inputs for the analyses, the actual cost data, representing twenty-seven program development efforts, are included.

DD FORM 1 JAN 64 1473

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
PROGRAMMING (COMPUTERS) COST DATA MULTIVARIATE REGRESSION ANALYSIS COST-ESTIMATING EQUATIONS FACTOR ANALYSIS						